



TUGAS AKHIR – TE 141599

**RANCANG BANGUN SISTEM PELACAKAN
OTOMATIS DAN PENGUNCIAN SASARAN PADA
PERTAHANAN STATIS BERBASIS PENGOLAH
CITRA**

Dani Prasetyawan
NRP 2211100011

Dosen Pembimbing
Rudy Dikairono, ST., MT.
Ronny Mardiyanto, ST., MT., Ph.D.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



TUGAS AKHIR – TE 141599

**RANCANG BANGUN SISTEM PELACAKAN
OTOMATIS DAN PENGUNCIAN SASARAN PADA
PERTAHANAN STATIS BERBASIS PENGOLAH
CITRA**

Dani Prasetyawan
NRP 2211100011

Dosen Pembimbing
Rudy Dikairono, ST., MT.
Ronny Mardiyanto, ST., MT., Ph.D.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE 141599

**DESIGN AND REALIZATION OF AUTO TRACKING AND
TARGET LOCK ON STATIC DEFENSE BASED ON IMAGE
PROCESSING**

Dani Prasetyawan
NRP 2211100011

Supervisor
Rudy Dikairono, ST., MT.
Ronny Mardiyanto, ST., MT., Ph.D.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2015

**RANCANG BANGUN SISTEM PELACAKAN
OTOMATIS DAN PENGUNCIAN SASARAN PADA
PERTAHANAN STATIS BERBASIS PENGOLAH
CITRA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

Bidang Studi Elektronika

Jurusan Teknik Elektro

Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I



Rudy Dikairono, ST., MT.
NIP. 198103252005011002

Dosen Pembimbing II



Ronny Mardiyanto, ST., MT., Ph.D.
NIP. 198101182003121003



RANCANG BANGUN SISTEM PELACAKAN OTOMATIS DAN PENGUNCIAN SASARAN PADA PERTAHANAN STATIS BERBASIS PENGOLAH CITRA

Nama : Dani Prasetyawan
Pembimbing I : Rudy Dikairono, ST., MT.
Pembimbing II : Ronny Mardiyanto, ST., MT, Ph.D.

ABSTRAK

Sistem pelacakan otomatis dan penguncian sasaran merupakan sebuah sistem senjata yang diotomasi. Sistem tersebut bekerja dengan melacak dan mengunci sasaran secara otomatis terhadap target yang telah dipilih terlebih dahulu. Sistem ini dapat menggantikan peran manusia pada suatu pertahanan titik. Sistem seperti ini yang ada sebelumnya menggunakan teknologi radar dan *opto-electrical*. Teknologi tersebut rentan *jamming* dan memiliki tingkat kesusahan yang tinggi serta biaya pengadaan yang mahal sehingga digunakanlah sistem serupa dengan basis penginderaan visual menggunakan kamera.

Kamera diletakkan pada sistem ini bertujuan untuk proses pengambilan data *visual*. Dari data *visual* tersebut dipilih target untuk diproses. Kemudian data yang didapat diproses oleh *processing unit* dengan tujuan untuk dapat melacak sasaran dan mengunci sasaran yang telah dipilih. Berikutnya *operator* dapat mengambil keputusan apakah sasaran perlu ditembak atau dibiarkan.

Hasil dari pengujian yang dilakukan pada tugas akhir ini adalah sistem dapat melacak sasaran berupa manusia dengan sasaran dapat berupa sasaran bergerak maupun sasaran diam yang dapat ditentukan operator. Lama waktu pelacakan dapat melebihi 40detik untuk sasaran bergerak. Untuk *error* penguncian sasaran adalah 10% dari lebar total *frame image* baik untuk sumbu tinggi maupun lebar. Selain itu rata-rata akurasi penembakan yang didapat dari pengujian penembakan dengan jarak dan ketinggian sasaran yang bervariasi adalah 50%.

Kata kunci : Kamera, penginderaan visual, *processing unit*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas kasih dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir ini dengan judul :

Rancang Bangun Sistem Pelacakan Otomatis dan Penguncian Sasaran Pada Pertahanan Statis Berbasis Pengolah Citra

Tugas Akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dibuat berdasarkan teori-teori yang didapat selama mengikuti perkuliahan, berbagai literatur penunjang dan pengarahan dosen pembimbing dari awal hingga akhir pengerjaan Tugas Akhir ini.

Pada kesempatan ini, penulis ingin berterima kasih kepada pihak-pihak yang membantu pembuatan tugas akhir ini, khususnya kepada:

1. Bapak, Ibu, adik serta seluruh keluarga yang memberikan dukungan baik moril maupun materiil.
2. Rudy Dikairono S.T., M.T. selaku dosen pembimbing 1 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
3. Ronny Mardiyanto, S.T., M.T.,P.hd. selaku dosen pembimbing 2 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
4. Tasripan, IR. MT selaku Koordinator Bidang Studi Elektronika.
5. Dr. Tri Arief Sardjono, ST, MT selaku Ketua Jurusan Teknik Elektro ITS Surabaya.
6. Seluruh dosen bidang studi elektronika.
7. Teman-teman laboratorium Elektronika yang tidak dapat disebutkan satu-persatu, telah membantu proses pengerjaan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini belum sempurna dan masih banyak hal yang dapat diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis untuk pengembangan lebih lanjut.

Akhirnya penulis berharap agar Tugas Akhir ini dapat memberikan manfaat yang sebesar-besarnya bagi semua pihak serta pengembangan aplikasi *image processing*. Penulis juga berharap supaya Tugas Akhir ini dapat menjadi aplikasi yang lebih bermanfaat.

Surabaya, 11 Mei
2015

Penulis

DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi	3
1.6 Sistematika Penulisan.....	5
1.7 Relevansi	5
BAB II TINJAUAN PUSTAKA DAN TEORI PENUNJANG	7
2.1 Citra Digital.....	7
2.2.1 Citra Biner.....	8
2.2.2 Citra Warna (RGB)	9
2.2.3 <i>Grayscale</i>	10
2.2 <i>Image Processing</i>	12
2.2.1 <i>Optical Flow</i>	12
2.2.1.1 Gambaran <i>Optical Flow</i>	12
2.2.1.2 Algoritma <i>Lucas-Kanade</i>	12
2.2.1.3 Pyramidal <i>Lukas-Kanade</i>	12
2.2.2 <i>Template Matching</i>	13
2.2.3 <i>Human Detection</i>	14
2.2.4 Emgu CV	14
2.2.5 Open CV	15
2.3 Jenis Senjata	15
2.3.1 <i>Machine Gun</i>	15
2.3.2 Submachine Gun	16
2.3.3 <i>Assault Rifle</i>	16
2.4 Data Acquisition Module.....	17

2.5	Pulse Width Modulation.....	18
2.6	Driver Motor.....	18
2.7	Gerak Peluru.....	20
2.8	Penelitian Terkait.....	21
2.8.1	CIWS (Close In Weapon System).....	21
BAB III PERANCANGAN SISTEM.....		23
3.1	Perancangan <i>Software</i>	27
3.1.1	Pengambilan Citra	27
3.1.2	<i>Grayscale</i>	28
3.1.3	Optical Flow	30
3.1.4	<i>Haar cascade</i>	32
3.1.5	Kontrol PID	33
3.1.6	Koneksi <i>Advantech DAQ Module</i>	34
3.2	Hardware	34
3.2.1	<i>Advantech USB-4711A</i>	34
3.2.2	Sensor <i>Proximity</i>	36
3.2.3	Rangkaian Pembangkit PWM (<i>Pulse Width Modulation</i>).....	37
3.2.4	Aktuator Tembak.....	40
3.2.5	<i>Driver Motor</i>	41
3.2.6	Motor DC	43
3.2.7	Senjata	44
3.2.8	Kamera	45
3.2.9	Komputer.....	45
3.2.10	Sistem Penembak.....	46
BAB IV PENGUJIAN.....		49
4.1	Pengujian Rangkaian Pembangkit PWM.....	49
4.2	Pengujian Motor	53
4.3	Pengujian Pendeteksi Manusia	54
4.4	Pengujian Pelacakan dan Penguncian Sasaran pada Siang Hari	57
4.5	.Pengujian Pelacakan dan Penguncian Sasaran di Malam Hari	67
4.6	Pengujian Penembakan.....	74
BAB V KESIMPULAN DAN SARAN.....		81
	Kesimpulan.....	81
	Saran.....	81

DAFTAR PUSTAKA	83
LAMPIRAN.....	85
BIODATA PENULIS.....	103

Halaman ini sengaja dikosongkan

DAFTAR TABEL

	Halaman
Tabel 3.1 Fungsi Penggunaan Port Advantech USB-4711	36
Tabel 3.2 Tabel kebenaran driver motor Vertikal.....	42
Tabel 3.3 Tabel kebenaran driver motor Horizontal.....	42
Tabel 4.1 Pengukuran keluaran <i>Duty Cycle</i> terhadap tegangan referensi	50
Tabel 4.2 Pengujian motor DC vertikal	53
Tabel 4.3 Pengujian motor DC horizontal	54
Tabel 4.4 Tabel pengujian deteksi manusia.....	55
Tabel 4.5 Tabel pengujian pelacakan pada siang hari	58
Tabel 4.6 Tabel alur gerak sasaran dan keterangan pengujian pelacakan pada siang hari.....	58
Tabel 4.7 Tabel pengujian pelacakan kedua pada siang hari purwarupa SIG552	61
Tabel 4.8 Tabel pengujian pelacakan kedua pada siang hari purwarupa M4	61
Tabel 4.9 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada siang hari purwarupa SIG552	62
Tabel 4.10 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada siang hari purwarupa M4.....	63
Tabel 4.11 Skenario pengujian gangguan sasaran	65
Tabel 4.12 Tabel pengujian pelacakan pada malam hari 1	67
Tabel 4.13 Tabel alur gerak sasaran dan keterangan pengujian pelacakan pada siang hari 1.....	68
Tabel 4.14 Tabel pengujian pelacakan kedua pada malam hari purwarupa SIG552	70
Tabel 4.15 T abel pengujian pelacakan kedua pada malam hari purwarupa M4	70
Tabel 4.16 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada malam hari purwarupa SIG552	71
Tabel 4.17 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada malam hari purwarupa M4	72
Tabel 4.18 Tabel pengujian tembak.....	75
Tabel 4.19 Tabel hasil pengujian tembak	75

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Halaman

Gambar 2.1 Posisi koordinat digital	8
Gambar 2.2 Citra Biner.....	9
Gambar 2.3 Representasi warna RGB pada citra digital	10
Gambar 2.4 Citra hasil operasi <i>grayscale</i>	11
Gambar 2.5 <i>Pyramidal Optical Flow</i>	13
Gambar 2.6 Logo EmguCV	14
Gambar 2.7 Logo OpenCV[8]	15
Gambar 2.8 M249 <i>Light Machine Gun</i>	16
Gambar 2.9 MP5 <i>Submachine Gun</i>	16
Gambar 2.10 M4 <i>Assault Rifle</i>	17
Gambar 2.11 Advantech Portable Data Acquisition Module [9].....	17
Gambar 2.12 Presentase <i>Duty Cycle</i> pada PWM.....	18
Gambar 2.13 Konfigurasi rangkaian <i>H-Bridge</i>	19
Gambar 2.14 Lintasan gerak peluru.....	20
Gambar 2.15 Phalanx CIWS[11].....	21
Gambar 3.1 Skenario pemilihan target	23
Gambar 3.2 Skenario sistem penembak mengikuti pergerakan target	24
Gambar 3.3 Diagram Blok Sistem.....	25
Gambar 3.4 <i>Flowchart</i> Sistem.....	26
Gambar 3.5 Tampilan <i>software</i>	27
Gambar 3.6 Citra frame	28
Gambar 3.7 Video hasil <i>capture</i> dalam RGB diubah ke <i>Grayscale</i>	28
Gambar 3.8 <i>Flowchart grayscale</i>	29
Gambar 3.9 Penandaan sasaran oleh program	31
Gambar 3.10 Diagram blok sistem kontrol PID	34
Gambar 3.11 Konfigurasi port <i>Advantech USB-4711A</i>	35
Gambar 3.12 Advantech USB-4711A	36
Gambar 3.13 Sensor <i>Proximity</i>	37
Gambar 3.14 Skematik rangkaian pembangkit PWM	38
Gambar 3.15 Rangkaian pembangkit PWM (dalam kotak).....	39
Gambar 3.16 Skematik rangkaian aktuator tembak	40
Gambar 3.17 Rangkaian aktuator menembak	41

Gambar 3.18 Driver motor BTN7870B.....	42
Gambar 3.19 Motor DC penggerak vertikal	43
Gambar 3.20 Motor DC penggerak horizontal	43
Gambar 3.21 Purwarupa M4 1:1.....	44
Gambar 3.22 Purwarupa SIG552 1:2.....	44
Gambar 3.23 Logitech C920.....	45
Gambar 3.24 Rancangan sistem penembak dengan purwarupa SIG552	46
Gambar 3.25 Rancangan sistem penembak dengan purwarupa M4	47
Gambar 3.26 Posisi motor DC penggerak horizontal	48
Gambar 4.1 Pengujian rangkaian pembangkit PWM.....	49
Gambar 4.2 Grafik hasil pengukuran keluaran <i>duty cycle</i> terhadap tegangan referensi	51
Gambar 4.3 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 0,8Volt.....	52
Gambar 4.4 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 2,2Volt.....	52
Gambar 4.5 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 3,6Volt.....	52
Gambar 4.6 Pengujian pendeteksi manusia	55
Gambar 4.7 Pengujian deteksi manusia	56
Gambar 4.8 Pengujian deteksi manusia	56
Gambar 4.9 Pengujian deteksi manusia	57
Gambar 4.10 Proses pengujian pelacakan pada siang hari	60
Gambar 4.11 Pengujian gangguan di depan sasaran sebelum gangguan	66
Gambar 4.12 Pengujian gangguan di depan sasaran sesudah gangguan menutupi sasaran	66
Gambar 4.13 Proses pengujian pelacakan pada malam hari	73
Gambar 4.14 Grafik Akurasi Penembakan Sasaran.....	78
Gambar 4.15 Proses pengujian penembakan	79

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam bidang militer, penggunaan tenaga manusia untuk melakukan tugas yang membutuhkan tingkat akurasi dan presisi yang tinggi perlahan dikurangi dan digantikan dengan sistem yang dapat berjalan secara mandiri. Contoh dari hal yang membutuhkan akurasi dan presisi yang tinggi adalah penajakan terhadap target dan penguncian terhadap target.

Seiring berkembangnya teknologi senjata, senjata yang digunakan semakin canggih, semakin cepat, dan sulit untuk dideteksi dan dikunci secara cepat dan akurat dengan kemampuan manusia. Manusia memiliki batasan untuk melihat benda dengan kecepatan tertentu dan pada jarak tertentu, sehingga membutuhkan bantuan alat untuk dapat menutupi kekurangan itu. Disamping itu, kemampuan manusia juga memiliki batasan berupa tidak dapat fokus pada suatu hal secara terus menerus diakibatkan memiliki batasan tenaga dan kelelahan akibat dari konsentrasi secara berkala. Hal tersebut dapat mengakibatkan kesalahan dalam pengambilan keputusan berupa keputusan untuk melakukan penembakan yang tidak tepat sasaran terhadap target yang telah dipilih. Hal ini sangat merugikan terutama untuk bidang militer.

Baru-baru ini senjata yang telah diotomasi pada beberapa aspek sedang sangat diminati karena dapat mencegahnya terjadinya *human error* maupun korban manusia dan dapat diandalkan. Senjata model seperti ini juga dapat mencegah atau mengurangi terjadinya korban yang tidak diinginkan akibat terjadinya *human error* karena tidak membutuhkan *operator* yang bekerja di tempat dan mengurangi tingkat akumulasi kelelahan pada *operator*.

Selain itu peperangan asimetris akhir-akhir ini menyebabkan banyak korban. Contohnya adalah bom bunuh diri yang ditempel pada tubuh pegebom bunuh diri menjadi salah satu penyumbang korban perang yang besar. Berdasar data oleh Dr Madelyn Hsiao-Rei Hicks, Institute of Psychiatry, King's College London, UK, korban bom bunuh diri di Iraq antara tahun 2003 hingga 2010 mencapai lebih dari 12.000 orang sipil, dan 300 personel militer.

Oleh karena itu dalam tugas akhir ini diimplementasikan rancang bangun sistem pelacakan otomatis dan penguncian sasaran untuk pertahanan statis berbasis pengolah citra. Dengan menggunakan teknik pengolah citra (*Image Processing*) yang dilakukan oleh *Processing Unit* dapat dilakukan penjajakan dan penguncian sasaran secara otomatis. Diharapkan agar sistem ini dapat menutupi kekurangan yang ada pada manusia untuk melakukan tugas serupa dan dapat mencegah timbulnya korban yang tidak diinginkan maupun terjadinya kesalahan yang terjadi akibat keterbatasan yang dimiliki oleh manusia.

1.2 Perumusan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini adalah:

1. Bagaimana mendeteksi target yang ada?
2. Bagaimana pelacakan target dapat dilakukan?
3. Bagaimana penguncian target dapat dilakukan?
4. Bagaimana merencanakan pergerakan sistem untuk mengikuti pergerakan target?

1.3 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah:

1. Pergerakan motor horizontal terdapat delay akibat sistem mekanik.
2. Motor vertikal terdapat tambahan beban akibat gaya gravitasi.
3. Lokasi pelaksanaan memiliki pencahayaan yang cukup.
4. Akurasi dan jarak penembakan bergantung jenis senjata.
5. Jarak deteksi minimal 3 Meter.
6. Target dipilih secara manual.

1.4 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Sistem mampu melacak target secara tepat.
2. Sistem mampu mengikuti pergerakan target secara tepat.
3. Sistem mampu bekerja dengan kecepatan yang bervariasi.
4. Dapat mengkomunikasikan hasil pemrosesan data citra hasil pelacakan dan penguncian kepada *operator* secara baik.

5. Dapat menempatkan kamera, *processing unit*, dan sistem senjata secara baik agar tidak membebani *operator*.

1.5 Metodologi

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut :

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Tugas Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, dan artikel-artikel di internet dan forum-forum diskusi internet.

2. Perancangan Software

Perangkat lunak dirancang dengan pembuatan *source code* melakukan pelacakan target, penguncian target dan penembakan target. Untuk pemilihan sasaran, dilakukan oleh *operator* secara *manual*. Untuk pelacakan dan penguncian sasaran, dapat dilakukan dengan membandingkan *error* dari jarak target sekarang menuju titik tengah (titik tembak) dan *error* pada *frame* dari *image* yang diterima saat itu dengan *error* pada *frame* sebelumnya yang ditangkap oleh kamera yang digunakan sebagai referensi sasaran. Lalu dilakukan perhitungan agar sasaran selalu berada di tengah dari hasil *image* yang diterima oleh kamera. Untuk melakukannya, diperlukan *source code* yang akan digunakan untuk mengirim perintah pada *hardware* agar bergerak mengikuti sesuai arahan dari *software*. Terakhir, dibuat *code* agar jika sasaran melebihi batasan dari sistem yang dibuat, maka sistem akan otomatis berhenti mengejar sasaran.

3. Perancangan Hardware

Perancangan *hardware* pada tahap ini meliputi kamera, *data acquisition module* untuk menggerakkan *hardware* berupa motor untuk pergerakan *pan-tilt*, danudukan

senjata serta kamera. Motor, kamera, dan senjata akan dihubungkan dengan *processing unit* agar dapat dikendalikan pergerakannya. Untuk kedudukan sistem dibuat tetap, tidak bergerak.

4. Pengujian Sistem

Pengujian alat dilakukan untuk menentukan keandalan dari sistem yang telah dirancang. Pengujian dilakukan untuk melihat apakah *software* dan *hardware* dapat bekerja secara baik.

Pengujian sistem dilakukan dengan pelacakan dan penguncian target atau sasaran yang bergerak yang telah dipilih. Apakah sistem dapat mengikuti pergerakan target tersebut dengan cara selalu mengarahkan senjata menuju target atau tidak. Lalu dilakukan pengujian tembak apakah tembakan dari senjata yang diarahkan mengenai sasaran yang telah dikunci tersebut

5. Analisa

Analisa dilakukan terhadap hasil dari pengujian sehingga dapat ditentukan karakteristik pelacakan dan penguncian dari *software* dan *hardware* yang telah dibuat. Apabila karakteristik pelacakan dan penguncian dari *software* dan *hardware* yang telah dibuat masih belum sesuai, maka perlu dilakukan perancangan ulang pada sistem dan diuji kembali.

6. Penulisan Laporan Tugas Akhir

Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, tinjauan pustaka dan teori penunjang, perancangan sistem, pengujian, dan penutup.

1.6 Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

➤ BAB I : PENDAHULUAN

Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

➤ BAB II : TINJAUAN PUSTAKA DAN TEORI PENUNJANG

Bab ini menjelaskan tentang teori penunjang dan *literature* yang dibutuhkan dalam pengerjaan tugas akhir ini. Dasar teori yang menunjang meliputi citra digital, *haarcascade upper body detection*, *optical flow*, *Pulse Width Modulation*, *Data Acquisition Module*, *driver motor*, dan Motor DC. Bagian ini memaparkan mengenai beberapa teori penunjang dan beberapa literatur yang berguna bagi pembuatan Tugas Akhir ini.

➤ BAB III : PERANCANGAN SISTEM

Bab ini menjelaskan tentang perencanaan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk sistem pelacakan dan penguncian target secara otomatis.

➤ BAB IV : PENGUJIAN

Pada bab ini akan menjelaskan hasil uji coba sistem beserta analisisnya.

➤ BAB V : PENUTUP

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran-saran untuk pengembangan lebih lanjut.

1.7 Relevansi

Hasil yang diharapkan dari tugas akhir ini diharapkan mampu meringankan pekerjaan manusia khususnya pada bidang militer khususnya untuk pertahanan statis. Pengembangan lebih lanjut dari

sistem ini adalah penambahan penembakan otomatis agar sistem dapat bekerja tanpa *operator* sehingga korban pada pihak yang tidak diinginkan pada pihak *operator* dapat dihilangkan.

BAB II

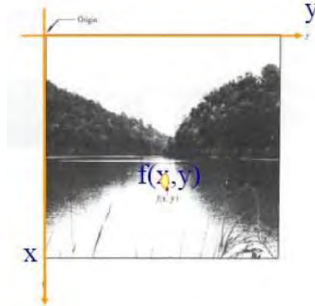
TINJAUAN PUSTAKA DAN TEORI PENUNJANG

Teori penunjang dalam bab ini menjelaskan tentang teori penunjang yang berhubungan dengan keseluruhan sistem yang akan dibuat pada tugas akhir ini. Sedangkan tinjauan pustaka dalam bab ini menjelaskan tentang sistem-sistem yang berhubungan dengan tugas akhir ini dan pernah diimplementasikan oleh penulis-penulis sebelumnya.

2.1 Citra Digital

Citra digital adalah gambar dua dimensi yang dapat ditampilkan pada layar monitor komputer sebagai himpunan berhingga (diskrit) dengan nilai digital yang disebut dengan *pixel* (*picture elements*). *Pixel* adalah elemen citra yang memiliki nilai yang menunjukkan intensitas warna. Citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar/ pixel/ piksel/ pels/ picture element) menyatakan tingkat keabuan pada titik tersebut. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi.

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial pada citra tersebut dan nilai $f(x,y)$ yang merupakan intensitas citra pada posisi koordinat tersebut. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*) dengan perubahan intensitas dari ketiga warna dasar tersebut menghasilkan intensitas pewarnaan yang berbeda. Sistem koordinat yang ada pada sebuah citra digital dapat dilihat pada contoh Gambar 2.1[1] berikut.



Gambar 2.1 Posisi koordinat digital

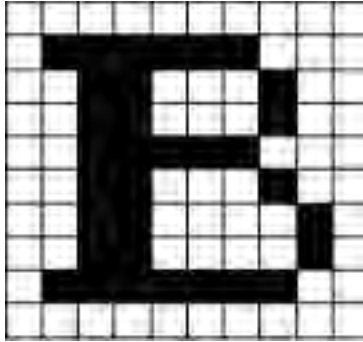
Berdasarkan cara penyimpanan atau pembentukannya, citra digital dapat dibagi menjadi dua jenis. Jenis pertama adalah citra digital yang dibentuk oleh kumpulan pixel dalam *array* dua dimensi. Citra jenis ini disebut citra bitmap (*bitmap image*) atau citra raster (*raster image*). Jenis citra yang kedua adalah citra yang dibentuk oleh fungsi-fungsi geometri dan matematika. Jenis citra ini disebut grafik vektor (*vector graphics*).

Citra digital (diskrit) dihasilkan dari citra analog (kontinu) melalui digitalisasi. Digitalisasi citra analog terdiri atas *sampling* dan kuantisasi (*quantization*). *Sampling* adalah pembagian citra ke dalam elemen-elemen diskrit (pixel), sedangkan kuantisasi adalah pemberian nilai intensitas warna pada setiap pixel dengan nilai yang berupa bilangan bulat (G.W. Awcock, 1996).

Banyaknya nilai yang dapat digunakan dalam kuantisasi citra bergantung kepada kedalaman pixel, yaitu banyaknya bit yang digunakan untuk merepresentasikan intensitas warna pixel. Kedalaman pixel sering disebut juga kedalaman warna. Citra digital yang memiliki kedalaman pixel n bit disebut juga citra n -bit.[2]

2.2.1 Citra Biner

Citra biner adalah citra dimana piksel-pikselnya hanya memiliki dua buah nilai intensitas yaitu bernilai 0 dan 1 dimana 0 menyatakan warna latar belakang (*background*) dan 1 menyatakan warna tinta/objek atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih. Citra biner diperoleh dari nilai citra *threshold* sebelumnya. Gradasi citra biner dapat dilihat pada Gambar 2.2. [3]



Gambar 2.2 Citra Biner

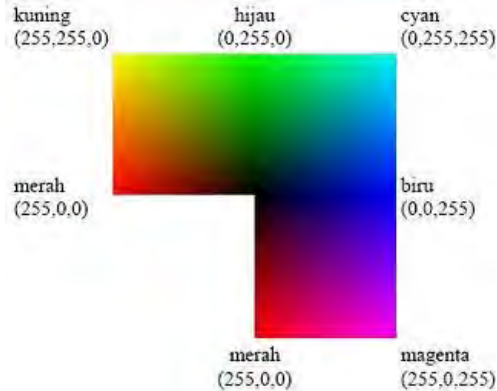
Meskipun saat ini citra berwarna lebih disukai karena memberi kesan warna yang lebih kaya dari pada citra biner, namun tidak membuat eksistensi citra biner mati. Pada beberapa aplikasi citra biner masih tetap di butuhkan, misalkan citra logo instansi (yang hanya terdiri dari warna hitam dan putih), citra kode barang (*bar code*) yang tertera pada label barang, citra hasil pemindaian dokumen teks, dan sebagainya. Seperti yang sudah disebutkan diatas, citra biner hanya mempunyai dua nilai derajat keabuan, hitam dan putih. Pixel – pixel objek bernilai 1 dan pixel – pixel latar belakang bernilai 0. Pada waktu menampilkan gambar, adalah putih dan 1 adalah hitam. Jadi pada citra biner, latar belakang berwarna putih sedangkan objek berwarna hitam seperti tampak pada gambar 2.2 diatas. Meskipun komputer saat ini dapat memproses citra hitam-putih (*grayscale*) maupun citra berwarna, namun citra biner masih tetap di pertahankan keberadaannya.

Alasan penggunaan citra biner adalah karena citra biner memiliki sejumlah keuntungan seperti kebutuhan memori kecil karena nilai derajat keabuan hanya membutuhkan representasi 1 bit dan Waktu pemrosesan lebih cepat di bandingkan dengan citra hitam-putih ataupun warna.

2.2.2 Citra Warna (RGB)

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang nilai. Untuk

monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak $256 \times 256 \times 256 = 1677726$ jenis warna.



Gambar 2.3 Representasi warna RGB pada citra digital

Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang dimensi 3 yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalkan sebuah vektor dituliskan sebagai $r = (x,y,z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen *Red*, *Green*, *Blue*. Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = $RGB(30, 75, 255)$. Putih = $RGB(255,255,255)$, sedangkan untuk hitam = $RGB(0,0,0)$. [4]

2.2.3 Grayscale

Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih" saja. Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra *grayscale* seringkali merupakan

perhiungan dari intensitas cahaya pada setiap pixel pada spektrum elektromagnetik single band.

Citra *grayscale* disimpan dalam format 8 bit untuk setiap sample piksel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi citra *grayscale* dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B. [5]

Citra *grayscale* merupakan citra satu kanal, dimana citra $f(x,y)$ merupakan fungsi tingkat keabuan dari hitam keputih, x menyatakan variable kolom atau posisi pixel di garis jelajah dan y menyatakan variable kolom atau posisi pixel di garis jelajah. Intensitas f dari gambar hitam putih pada titik (x,y) disebut derajat keabuan (*grey level*), yang dalam hal ini derajat keabuannya bergerak dari hitam keputih. Derajat keabuan memiliki rentang nilai dari I_{min} sampai I_{max} , atau $I_{min} < f < I_{max}$, selang (I_{min} , I_{max}) disebut skala keabuan.

Biasanya selang (I_{min} , I_{max}) sering digeser untuk alasan-alasan praktis menjadi selang $[0,L]$, yang dalam hal ini nilai intensitas 0 menyatakan hitam, nilai intensitas L menyatakan putih, sedangkan nilai intensitas antara 0 sampai L bergeser dari hitam ke putih. Sebagai contoh citra *grayscale* dengan 256 level artinya mempunyai skala abu dari 0 sampai 255 atau $[0,255]$, yang dalam hal ini intensitas 0 menyatakan hitam, intensitas 255 menyatakan putih, dan nilai antara 0 sampai 255 menyatakan warna keabuan yang terletak antara hitam dan putih.



Gambar 2.4 Citra hasil operasi *grayscale*

2.2 Image Processing

2.2.1 Optical Flow

2.2.1.1 Gambaran Optical Flow

Optical Flow adalah perkiraan gerakan suatu bagian dari sebuah citra berdasarkan turunan intensitas cahayanya pada sebuah sekuen citra. Pada ruang 2D hal ini berarti seberapa jauh suatu piksel citra berpindah diantara dua *frame* citra yang berurutan. Sedangkan pada ruang 3D hal ini berarti seberapa jauh suatu *volume* pixel (*voxel*) berpindah pada dua *volume* yang berurutan. Perhitungan turunan dilakukan berdasarkan perubahan intensitas cahaya pada kedua *frame* citra maupun *volume*. Perubahan intensitas cahaya pada suatu bagian citra dapat disebabkan oleh gerakan yang dilakukan oleh obyek, gerakan sumber cahaya, ataupun perubahan sudut pandang.

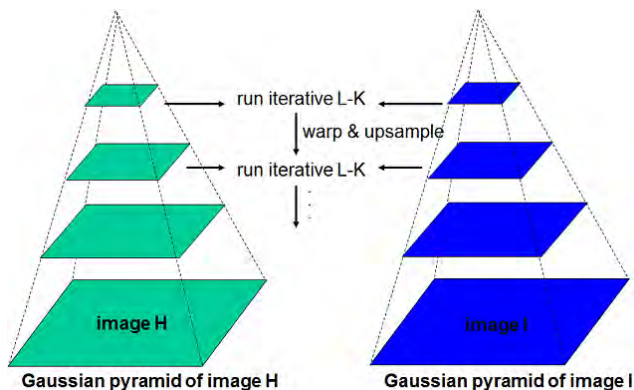
2.2.1.2 Algoritma Lucas-Kanade

Algoritma *Lucas-Kanade*, pertama kali diajukan pada tahun 1981. Pada awalnya algoritma ini adalah sebuah usaha untuk mencari teknik registrasi citra yang cepat dengan memanfaatkan *gradient* intensitas spasial. Pada perkembangannya, algoritma ini kemudian menjadi salah satu algoritma *optical flow* yang penting. Berbeda dengan algoritma *Horn-Schunk* yang bekerja berbasis pada keseluruhan citra, algoritma ini bekerja berdasar pada informasi lokal yang diturunkan dari *window* kecil (*patch*) disekeliling titik yang diperhitungkan dalam proses. Kelemahan digunakan *window* local kecil pada algoritma *Lucas-Kanade* adalah tidak terdeteksinya gerakan-gerakan yang besar karena gerakan-gerakan tersebut jatuh diluar *window*. Akibatnya terjadi *miss* pada gerakan yang besar. Permasalahan ini kemudian dapat diatasi dengan mengimplementasikan penyelesaian dengan prinsip piramida, yaitu *pyramidal Lucas-Kanade*. Prinsip ini merupakan penyelesaian berdasar iterasi dari level detail citra paling rendah hingga *level* detail citra paling tinggi.

2.2.1.3 Pyramidal Lukas-Kanade

Penyelesaian algoritma *Lukas-Kanade* dengan pendekatan piramida, atau yang umum disebut dengan *Pyramidal Lukas-Kanade*

diajukan pertama kali oleh Jean-Yves (Bouquet, pada tahun 2000). Pendekatan ini menggunakan prinsip piramida, yaitu bekerja mulai dari detil citra paling rendah hingga detil citra paling tinggi yang diilustrasikan pada gambar 2.5. Tujuannya adalah agar gerakan-gerakan yang “besar” dapat ikut diperhitungkan. Sementara asumsi yang digunakan pada algoritma *Lucas-Kanade* adalah gerakan yang “kecil” dan koheren, sehingga tidak dapat menangkap gerakan yang “besar”. Solusi untuk dapat menangkap gerakan yang “besar” pada algoritma *Lucas-Kanade* adalah dengan menggunakan *window* yang besar. Tetapi, akibat dari penggunaan *window* yang besar sering kali membuat gerakan yang ditangkap adalah gerakan yang tidak koheren. Algoritma *Pyramidal Lukas-Kanade* dapat menyelesaikan permasalahan tersebut tanpa menghilangkan asumsi gerakan-gerakan yang koheren.



Gambar 2.5 *Pyramidal Optical Flow*

Algoritma *Pyramidal Lucas-Kanade* pertama bekerja pada layer piramida paling tinggi. Kemudian hasilnya digunakan sebagai titik awal untuk bekerja pada *layer* dibawahnya. Hal ini berlanjut hinggamencapai *level* paling rendah. [6]

2.2.2 *Template Matching*

Template matching adalah sebuah teknik dalam pengolahan citra digital untuk menemukan bagian-bagian kecil dari gambar yang cocok

dengan template gambar. Energi cahaya yang terpancar dari suatu bentuk mengenai retina mata dan diubah menjadi energi neural yang kemudian dikirim ke otak. Selanjutnya terjadi pencarian di antara template - template yang ada. Jika sebuah template ditemukan sesuai (*match*) dengan pola tadi, maka subjek dapat mengenal bentuk tersebut. Setelah kecocokan antara objek dan *template* terjadi, proses lebih lanjut dan interpretasi terhadap objek bisa terjadi. [7]

2.2.3 *Human Detection*

Dalam proses pengenalan target, untuk mengetahui ada mengetahui apa saja yang ada di depan *user* akan sangat membantu *user* dalam berinteraksi. *Human Detection* dipilih sebagai salah satu fitur yang diberikan untuk membantu *user*. *Human Detection* adalah sebuah metode untuk mendeteksi ada atau tidaknya orang yang berada di depan *user*. Metode ini memanfaatkan library tambahan yang ada pada *emgucv* yaitu “*haarcascade_upperbody.xml*”. Metode ini mendeteksi bagian atas dari manusia yang nantinya akan diproses dan dikomunikasikan kepada *user* apabila terdeteksi manusia yang berada di depannya.

2.2.4. *Emgu CV*

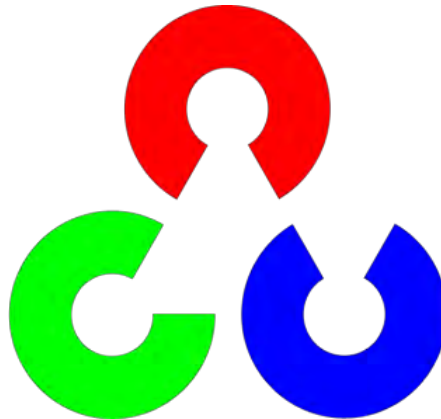
EmguCV adalah wrapper .Net untuk *OpenCV*. Dengan *EmguCV*, fungsi-fungsi dalam *OpenCV* bisa dipanggil melalui bahasa pemrograman yang *compatible* dengan .NET seperti C#, VB, dan VC++. Keuntungan menggunakan *EmguCV* yang paling utama adalah *library* ini sepenuhnya ditulis dengan C# yang mana tentunya lebih *safe* karena pembuatan *object* atau pun *reference* di-manage oleh *garbage collector*. Selain itu, *Emgu CV* juga *cross platform* sehingga dapat di-*compile* lewat Mono dan dijalankan di atas sistem operasi Linux atau Mac OS.



Gambar 2.6 Logo *EmguCV*

2.2.5. Open CV

OpenCV (Open source Computer Vision) merupakan sebuah library dari fungsi programming untuk real-time computer vision. OpenCV dapat dimanfaatkan oleh program-program lainnya (seperti C++, C dan Python) untuk melakukan pengambilan, pengolahan serta penampilan data gambar, baik dalam bentuk image dan video maupun real-time video. OpenCV bersifat Open Source (dapat digunakan secara bebas) baik untuk akademik maupun untuk komersil, yang dapat berjalan dengan operating system Windows, Linux, Android dan Mac.



Gambar 2.7 Logo OpenCV[8]

2.3 Jenis Senjata

2.3.1 *Machine Gun*

Machine Gun (senapan mesin) adalah senjata api otomatis dipasang pada titik atau dibawa secara *portable* oleh setiap orang. Dirancang untuk menembak peluru secara berurutan dari sabuk amunisi atau magasin dan memiliki tingkat kecepatan menembak pada 3-1800 putaran per menit dan memiliki jarak efektif yang jauh bergantung dari kaliber yang digunakan.



Gambar 2.8 *M249 Light Machine Gun*

2.3.2 Submachine Gun

Submachine Gun adalah sebuah senjata api yang menggabungkan kemampuan menembak otomatis senapan mesin dengan amunisi pistol. Konsep senjata api seperti ini pertama kali dicoba pada tahun 1900-an, yaitu pistol yang diberi popor dan menembak secara otomatis.



Gambar 2.9 *MP5 Submachine Gun*

2.3.3 Assault Rifle

Assault Rifle (Senapan serbu) adalah senjata api otomatis yang merupakan senapan laras panjang atau karabin, yang memiliki pilihan tembakan (*selective-fire*), dan menggunakan amunisi kaliber menengah. Senapan serbu masuk dalam kategori di antara senapan mesin ringan, yang berfungsi untuk menembak secara *full*-otomatis sebagai senjata pendukung, dan *submachine gun*, senjata otomatis yang menggunakan peluru ukuran peluru pistol sebagai senjata api jarak dekat. Senapan

serbu sudah menjadi persenjataan standar untuk tentara modern, menggantikan senapan laras panjang era Perang Dunia II.



Gambar 2.10 M4 *Assault Rifle*

2.4 Data Acquisition Module

Data Acquisition Module adalah suatu *Module* yang memberikan hasil akuisisi data menuju computer. Sedangkan *Data Acquisition* adalah proses pengambilan sampel dari sebuah sinyal yang mengukur kondisi fisik dunia nyata dan mengkonversi sampel yang dihasilkan menjadi nilai numerik digital sehingga dapat diproses oleh komputer. Sistem akuisisi data biasanya mengkonversi bentuk gelombang analog menjadi nilai digital untuk diproses. Komponen sistem akuisisi data meliputi:

- Sensor yang mengkonversi parameter fisik menjadi sinyal-sinyal listrik.
- Sinyal pengkondisi rangkaian untuk mengkonversi sinyal sensor menjadi bentuk yang dapat dikonversi ke nilai digital.
- *Analog-to-digital converter*, yang mengkonversi sinyal sensor yang berupa analog menuju sinyal digital.

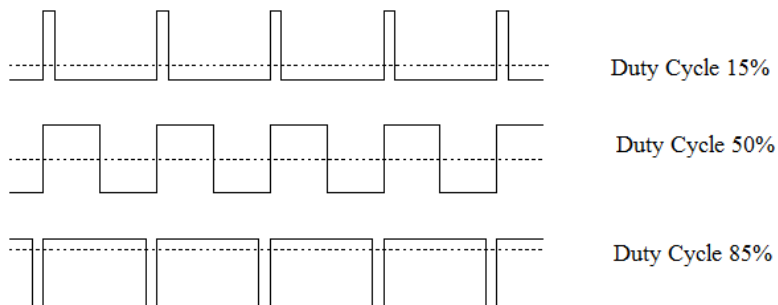


Gambar 2.11 Advantech Portable Data Acquisition Module [9]

2.5 Pulse Width Modulation

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu perioda, untuk mendapatkan tegangan rata-rata yang berbeda. Beberapa contoh aplikasi PWM adalah pemodulasian data untuk telekomunikasi, pengontrolan daya atau tegangan yang masuk ke beban, regulator tegangan, audio effect dan penguatan, serta aplikasi-aplikasi lainnya. Aplikasi PWM berbasis mikrokontroler biasanya berupa pengendalian kecepatan motor DC, pengendalian motor servo, pengaturan nyala terang LED dan lain sebagainya.

Sinyal PWM pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar Pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. Artinya, Sinyal PWM memiliki frekuensi gelombang yang tetap namun duty cycle bervariasi (antara 0% hingga 100%).[9]

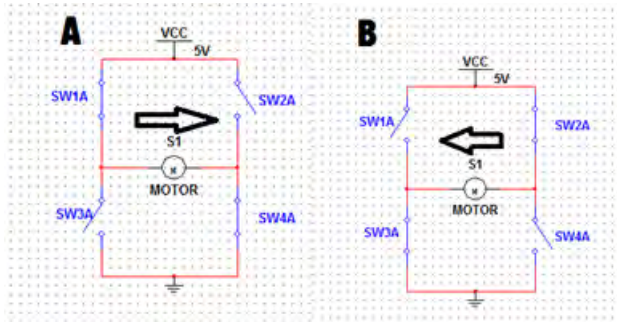


Gambar 2.12 Presentase *Duty Cycle* pada PWM

2.6 Driver Motor

Driver motor yang biasa digunakan adalah konfigurasi *H-Bridge* dimana arah pergerakan motor dapat dikendalikan sesuai dengan arah arum jam atau berlawanan dengan jarum jam. Pada prinsipnya, rangkaian ini terdiri dari empat buah saklar yang disusun sedemikian

rupa sehingga motor DC dapat dialiri arus dengan arah yang berkebalikan. [10]. Berikut gambar untuk menjelaskan prinsip tersebut:

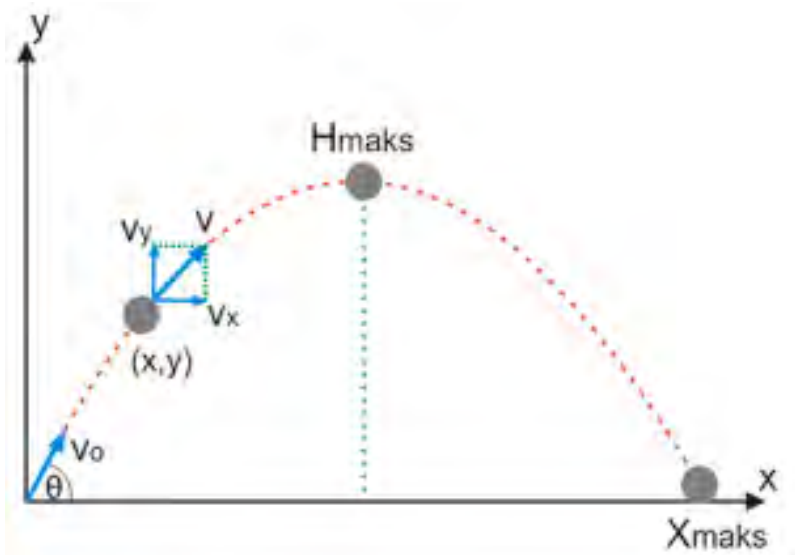


Gambar 2.13 Konfigurasi rangkaian *H-Bridge*

Prinsip kerja dari rangkaian H-bridge adalah mengatur arah alir arus pada motor dengan cara membuka atau menutup switch yang ada sehingga motor dapat diatur untuk bergerak searah jarum jam atau berlawanan dengan jarum jam. Ketika SW1A dan SW4A tertutup sedangkan SW2A dan SW3A terbuka (seperti pada gambar 2.11 A) maka sisi kiri dari motor akan tersambung dengan VCC dan sisi sebelah kanan motor akan terhubung dengan ground sehingga kondisi ini akan membuat arah pergerakan motor kekanan atau searah dengan jarum jam. Begitupun juga dengan kondisi sebaliknya, pada saat SW2A dan SW3A tertutup sedangkan SW1A dan SW4A terbuka (seperti pada gambar 2.11 B), maka arah alir arus akan bergerak dari sisi kanan motor menuju ke sisi kiri motor karena sisi kanan motor terhubung dengan VCC sedangkan sisi kiri motor terhubung dengan ground sehingga motor akan bergerak kearah kiri atau berlawanan dengan jarum jam. Pada penerapannya, switch ini biasanya diganti dengan saklar otomatis, salah satunya adalah dengan komponen transistor dimana untuk mengalirkan arus dari sisi kolektor kesisi emitor (*short*), kita hanya perlu memberikan arus forward pada basis, sehingga fungsi transistor tersebut akan berperan sebagai saklar.

2.7 Gerak Peluru

Gerak peluru atau parabola adalah perpaduan antara gerak horizontal (searah dengan sumbu x) dengan vertikal (searah sumbu y). Pada gerak horizontal bersifat GLB (Gerak Lurus Beraturan) karena gesekan udara diabaikan. Sedangkan pada gerak vertikal bersifat GLBB (Gerak Lurus Berubah Beraturan) karena pengaruh percepatan gravitasi bumi (g) terhadap kecepatan vertikal.



Gambar 2.14 Lintasan gerak peluru

Perumusan jarak maksimal gerak peluru (X_{maks}) dapat dihitung dengan:

$$X_{maks} = \frac{V_o^2 \cdot \sin^2 \alpha}{g} \quad (2-1)$$

Dari perumusan tersebut, gesekan udara dan pengaruh massa diabaikan. Maka kecepatan awal lekat benda mempengaruhi jarak maksimum tempuh benda.

2.8 Penelitian Terkait

2.8.1 CIWS (Close In Weapon System)

Close In Weapon System (CIWS), sering diucapkan *sea-whiz*, adalah senjata titik pertahanan point-defense untuk mendeteksi dan menghancurkan rudal jarak pendek yang masuk dan pesawat musuh yang telah menembus pertahanan luar, biasanya dipasang pada kapal dalam kapasitas angkatan laut.



Gambar 2.15 Phalanx CIWS[11]

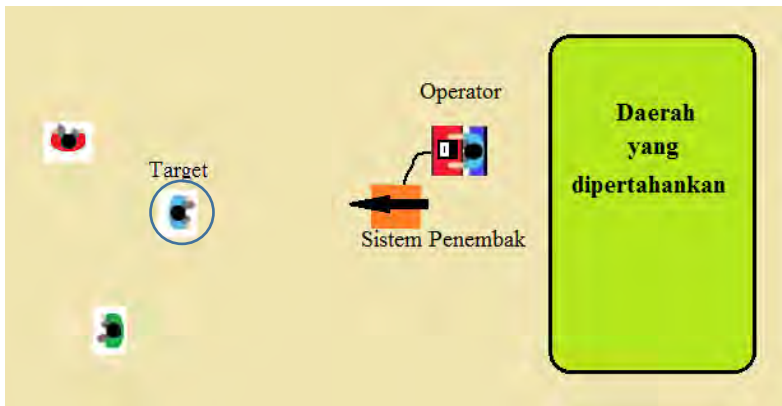
-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN SISTEM

Pada bab ini akan dibahas perancangan keseluruhan sistem dimulai dari penjelasan bagaimana skenario pelacakan dan penguncian target sampai pada perancangan *software* dan *hardware* dari sistem pelacakan dan penguncian target otomatis.

Pada tugas akhir ini memanfaatkan teknologi pengolah citra untuk melakukan pelacakan dan penguncian target secara otomatis. Sistem ini digunakan untuk mempertahankan suatu daerah dan berkonfigurasi pertahanan statis. Pada sistem ini *operator* memilih sasaran yang dirasa akan membahayakan dan sistem akan melacak dan mengunci sasaran yang telah dipilih oleh operator tersebut. Berikut adalah ilustrasi dan penjelasan peletakan sistem serta proses dari pemilihan sasaran, pelacakan, dan penguncian sasaran dilakukan.

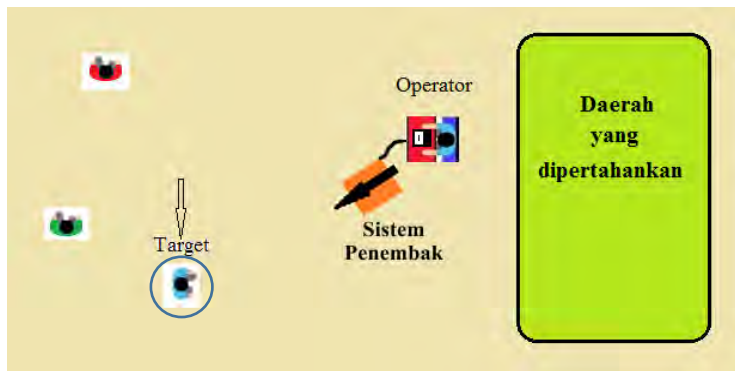


Gambar 3.1 Skenario pemilihan target

Sistem diletakkan untuk mempertahankan suatu *area* atau daerah yang dipertahankan. Sistem ini peletakannya statis sehingga disebut juga sebagai sistem pertahanan statis atau titik. Sasaran utama untuk sistem ini adalah manusia sehingga peletakan ketinggian sistem berada pada ketinggian 80cm hingga 150 cm. Sistem ini membutuhkan seorang

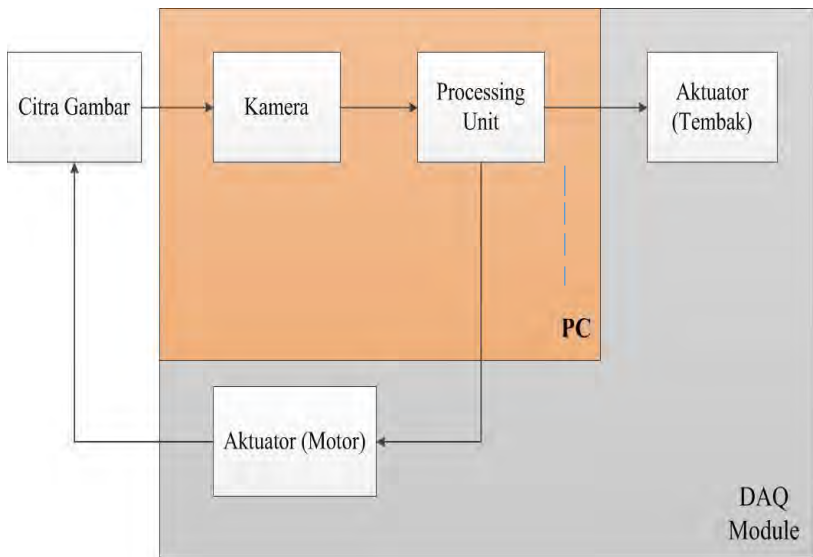
operator untuk mengoperasikannya. Saat dioperasikan, sistem akan menampilkan citra dan menampilkan keberadaan orang yang ada didepan sistem tersebut. Jika *operator* menganggap ada seseorang yang membahayakan, maka *operator* cukup melakukan pemilihan sasaran terhadap orang yang dicurigai lalu sistem akan mengikuti pergerakan sasaran yang telah dipilih. *Operator* dapat melakukan penggantian sasaran maupun menghapus sasaran yang telah dipilih. Jika sasaran dianggap perlu ditembak jatuh oleh *operator*, maka *operator* juga dapat memberikan perintah penembakan terhadap target melalui *processing unit* dan dapat memilih sasaran berikutnya jika diperlukan. Semua proses yang diperintahkan oleh *operator* dilakukan melalui *processing unit* atau komputer tanpa menyentuh sistem *hardware* (sistem penembak) secara langsung sehingga *operator* dapat berada pada jarak yang aman.

Pada gambar 3.1 diilustrasikan *operator* memilih orang berbaju biru sebagai sasaran atau target karena dianggap membahayakan atau mencurigakan. Sedangkan orang berbaju merah dan hijau dianggap tidak membahayakan sehingga tidak dimasukkan dalam sasaran oleh operator. Setelah pemilihan sasaran dilakukan, *processing unit* akan melacak pergerakan sasaran memerintahkan sistem penembak untuk mengikuti pergerakan dari sasaran tersebut. Berikut gambaran dari sistem penembak yang mengikuti pergerakan dari sasaran yang telah dipilih.

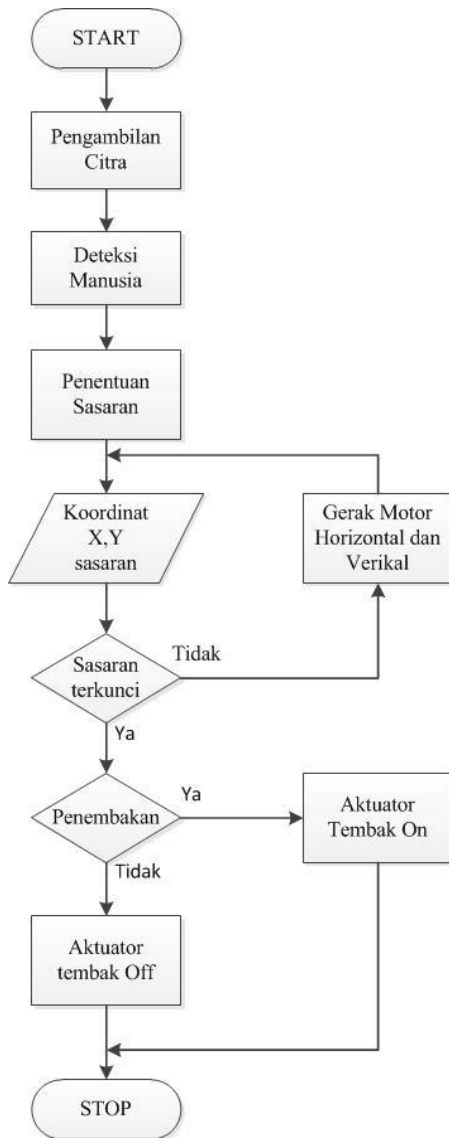


Gambar 3.2 Skenario sistem penembak mengikuti pergerakan target

Pada gambar 3.2 diilustrasikan sistem penembak akan tetap mengarah pada sasaran target berbaju biru yang pergerakannya tunjukkan oleh tanda panah. Sistem akan melacak dan mengunci sasaran yang telah dipilih ke manapun sasaran bergerak dalam *coverage area* sistem penembak. Terlihat pada gambar tersebut sistem penembak tetap mengarah menuju sasaran yang telah bergerak. Berikutnya jika oleh *operator* yang mengoperasikan sistem tersebut merasa target tersebut membahayakan dan perlu dilakukannya penembakan, maka *operator* dapat memerintahkan sistem penembak untuk menembak sasaran tersebut. Penembakan dapat dilakukan tanpa *operator* menyentuh sistem penembak secara langsung. *Operator* cukup memberikan perintah penembakan melalui *processing unit*. Berikut diagram blok sistem dan *flowchart* sistem secara keseluruhan.



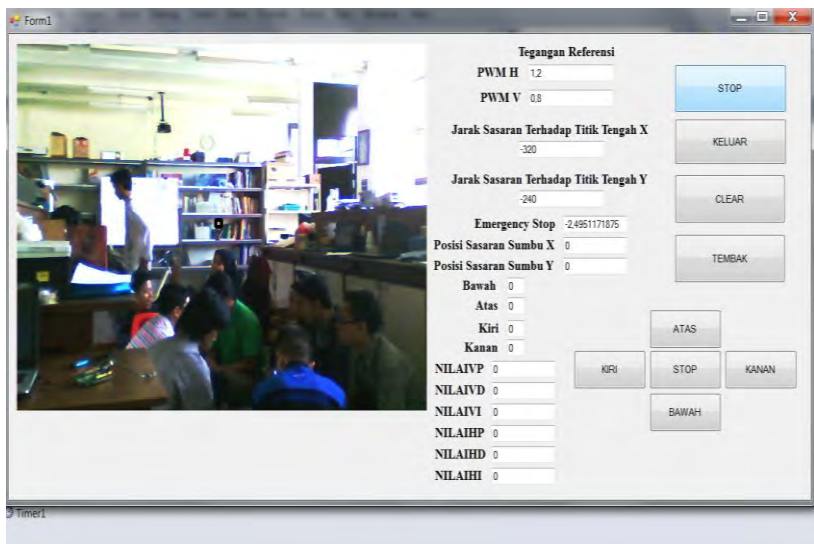
Gambar 3.3 Diagram Blok Sistem



Gambar 3.4 *Flowchart* Sistem

3.1 Perancangan Software

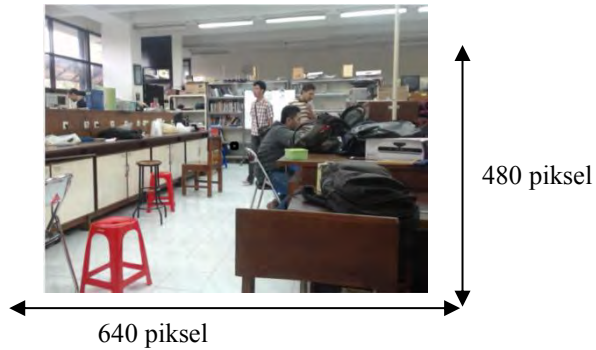
Tahapan ini dirancang algoritma pemrograman untuk melakukan proses berupa pengambilan citra, *grayscale*, *optical flow*, pemilihan sasaran, pergerakan manual, serta penembakan. Semua proses perangkat lunak dilakukan oleh *processing unit* atau komputer. Algoritma pemrograman ini digunakan untuk mengatur hardware berupa DAQ module dan mengatur pergerakan motor untuk gerak vertikal, horizontal, dan menembak.



Gambar 3.5 Tampilan software

3.1.1 Pengambilan Citra

Data citra yang didapat pada input citra dari kamera disimpan dalam bentuk variabel frame. Variabel inilah yang akan diolah dengan metode *optical flow* input citra. Pada sistem ini data citra yang disimpan memiliki ukuran 640x480 piksel. Semakin besar ukuran piksel yang disimpan, maka semakin besar proses yang terjadi sehingga dapat menyebabkan *delay* pada sistem.



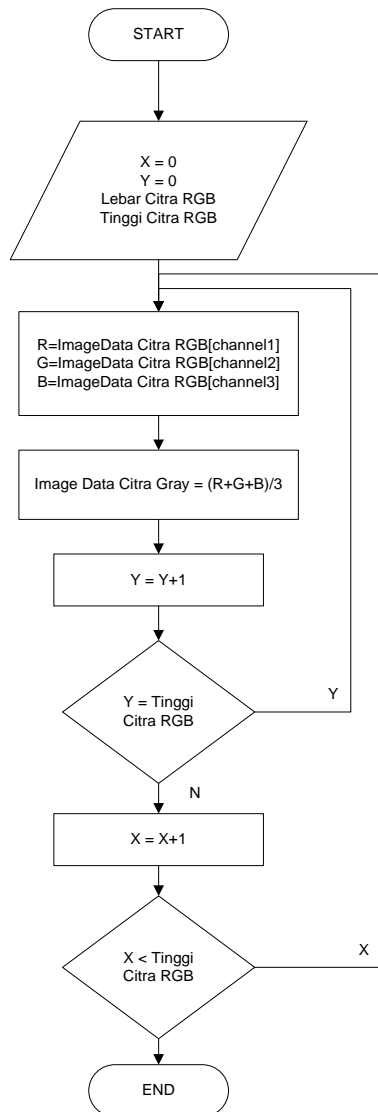
Gambar 3.6 Citra frame

3.1.2 *Grayscale*

Langkah pertama mengolah data citra untuk proses *optical flow* adalah dengan mengubah data citra awal yang bernilai RGB menjadi *grayscale*. Proses ini dilakukan karena *RGB* memiliki 24 bit, dimana masing-masing kanal warna memiliki nilai 8 bit yaitu 8 bit untuk *Red*, 8 bit untuk *Green*, dan 8 bit untuk *Blue*. Jika tidak dikonversi kedalam bentuk *grayscale*, maka perhitungan nilai piksel akan sangat sangat banyak sehingga memperlama proses. Maka perubahan ke *greyscale* membantu penghitungan untuk proses *optical flow* karena metode *surf* memerlukan frame yang bernilai 8 bit untuk mengurangi perhitungan yang kompleks.



Gambar 3.7 Video hasil *capture* dalam RGB diubah ke *Grayscale*



Gambar 3.8 *Flowchart grayscale*

Pada tugas akhir ini, digunakan fungsi yang sudah ada dalam library emgucv untuk mengubah dari citra RGB ke citra *grayscale*. berikut adalah prosedurnya :

```
Dim citraasli As New Image(Of Bgr, Byte)(640, 480)

frame = citraasli
grayB = frame.Convert(Of Emgu.CV.Structure.Gray,
Byte)()
```

citraasli : merupakan sumber frame yang akan dikonversi.
frame : merupakan wadah pengganti sumber frame yang akan di konversi.
grayB : merupakan frame yang akan dipakai sebagai wadah hasil konversi.
.Convert(Of Emgu.CV.Structure.Gray, Byte)()
: Perintah untuk mengkonversi dari RGB ke GRAY

Secara Manual Mengubah RGB kedalam bentuk Grayscale dalam dilakukan dengan cara menjumlah data gambar tiap kanal R,G, dan B dan dibagi 3. Hal tersebut dapat dilihat pada gambar 3.7 *flowchart grayscale* diatas.

$$Image\ data\ citra\ gray = \frac{R+G+B}{3} \quad (3-1)$$

3.1.3 Optical Flow

Metode *optical flow* merupakan metode yang menggunakan intensitas cahaya sebagai dasar untuk mendeteksi objek. Memiliki keuntungan karena menghasilkan vector gerakan dengan tingkat kepadatan yang tinggi. Pada tugas akhir ini, sasaran dipilih oleh *operator* dan sistem melakukan pelacakan dengan *optical flow* dengan referensi intensitas cahaya awal sasaran yang dipilih tersebut.

Dalam *optical flow*, referensi yang telah dipilih akan dihitung intensitas *brightness* dari referensi tersebut. Dengan asumsi referensi piksel di lokasi (x,y,t) dengan intensitas $I(x,y,t)$ melakukan perpindahan sebanyak Δx , Δy , dan Δt pada frame berikutnya, ketetapan *brightness* dapat diberikan:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (3-2)$$

Dalam program yang dibuat, sebelum dilakukan pelacakan sasaran, pemilihan titik awal sasaran dilakukan terlebih dahulu oleh *operator* dan akan ditandai oleh program. Berikut tampilan penandaan sasaran oleh program:



Gambar 3.9 Penandaan sasaran oleh program

Dalam gambar 3.8 diatas, sasaran yang diberi tanda adalah tanda bundar biru muda. Program akan secara otomatis mengikuti tanda yang telah ditandai kemanapun bergerak dalam frame. Dalam program yang

telah dibuat, referensi yang digunakan untuk dilacak hanya 1 tanda saja. Tanda lain yang diberikan berikutnya akan diabaikan oleh program.

3.1.4 Haar cascade

Dalam tugas akhir ini digunakan metode haar cascade, yang merupakan bagian dari metode *viola jones*. Metode *viola jones* adalah metode pendeteksi objek yang terdapat EmguCV. Metode ini merupakan metode yang paling banyak digunakan untuk mendeteksi objek, hal ini dikarenakan metode *viola jones* memiliki algoritma yang efisien, sehingga tidak memerlukan waktu lama dalam melakukan proses pendeteksian objek. Teori ini menjumlahkan nilai-nilai dari beberapa citra dalam sebuah kotak *frame*. Penjumlahan tersebut dapat dilihat dalam formula di bawah ini,

$$\sum_k \delta_k B_k(I) \quad (3-3)$$

dimana, $\delta \in \{1, -1\}$ dan

$$B_k(I) = \sum_{i=u1(k)}^{u2(k)} \sum_{j=v1(k)}^{v2(k)} I_y \quad (3-4)$$

Beberapa fitur dapat dengan cepat di evaluasi oleh integral *image*, dimana formula integral *image* dapat di bawah ini,

$$I_y = \sum_{u=1}^i \sum_{v=1}^j I_{uv} \quad (3-5)$$

dimana u dan v adalah nilai dari pixel integral *image*. Maka dari itu penjumlahan nilai dari sebuah *frame* dapat dievaluasi dengan 4 tahap integral *image*. Maka dengan mudah kita dapat memeriksa bahwa

$$\sum_{i=u1}^{u2} \sum_{j=v1}^{v2} I_y = I_{u2v2} - I_{u1v2} - I_{u2v1} + I_{u1v1} \quad (3-6)$$

setiap fitur dari citra yang tertangkap dapat dievaluasi dengan metode integral *image*[12]. Algoritma deteksi *upper body* dapat dilihat pada tabel di bawah ini

1. Deteksi *upper body*

- Membuat sebuah detektor objek *cascade*.
- Mengambil *frame image* dari *image* yang diambil dan jalankan detektor.

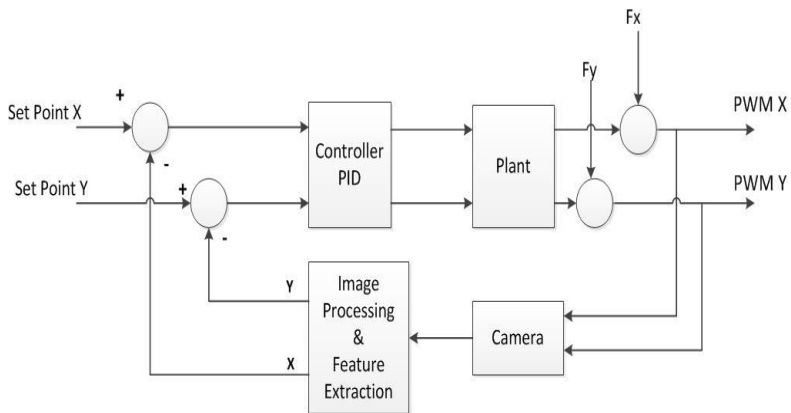
- Gambarkan sebuah kotak yang akan memberi tanda bila ada *upper body* yang terdeteksi.
- 2. Identifikasi fitur *upper body* untuk dilakukan pelacakan
 - Dapatkan informasi *upper body* dengan cara mengekstrak *Hue* dari setiap *frame image*.
 - Ubah citra tersebut ke dalam bentuk warna Hue, Saturation and Value (HSV).
 - Tampilkan *channel data Hue* dan gambar kotak di sekitar *upper body*.
 - Deteksi *upper body* di sekitar area *frame image*.
- 3. Pelacakan *upper body*
 - Buat sebuah pelacak objek.
 - Inisialisasi pelacak histogram menggunakan *channel pixel Hue*.
 - Buat sebuah *video* untuk menampilkan tiap *frame image*.
 - Lacak *upper body* di setiap *frame image* hingga pengambilan *frame* selesai
 - Beri sebuah kotak mengelilingi *upper body* pada objek yang terdeteksi
 - Tampilkan dalam *frame image* yang telah diolah

3.1.5 Kontrol PID

Pada *processing unit*, untuk mengatur agar gerakan motor dc yang digunakan penggerak vertikal dan horizontal pada sistem penembak diproses dengan control PID terlebih dahulu. Dalam metode pelacakan, selisih dari posisi piksel sasaran terhadap posisi posisi tengah frame (set point) untuk penguncian dianggap *error* untuk masukan E pada perhitungan kontrol PID oleh *processing unit* dan diperhitungkan seberapa besar kontrol pergerakan yang diperlukan untuk mengunci sasaran. Keluaran dari perhitungan ini adalah nilai arah gerak dan seberapa cepat motor perlu digerakkan.

Pada sistem ini yang digunakan adalah kontrol PID dengan bias dimana masukan bias (F_x untuk X dan F_y untuk Y) digunakan untuk mengatasi efek berat senjata ke motor horizontal maupun vertikal yang menyebabkan adanya zona mati dimana memerlukan *duty cycle* minimal

untuk menggerakkan motor (motor tidak langsung bergerak begitu *duty cycle* naik lebih dari 0%). Keluaran nilai PID pada sistem ini adalah antara 0 hingga 4 dimana akan direpresentasikan menjadi nilai tegangan analog yang akan keluar dari putput analog pada *DAQ Module* yang akan mengatur keluaran PWM pada rangkaian PWM. Dimana gambaran kontrol PID adalah sebagai berikut:



Gambar 3.10 Diagram blok sistem kontrol PID

3.1.6 Koneksi Advantech DAQ Module

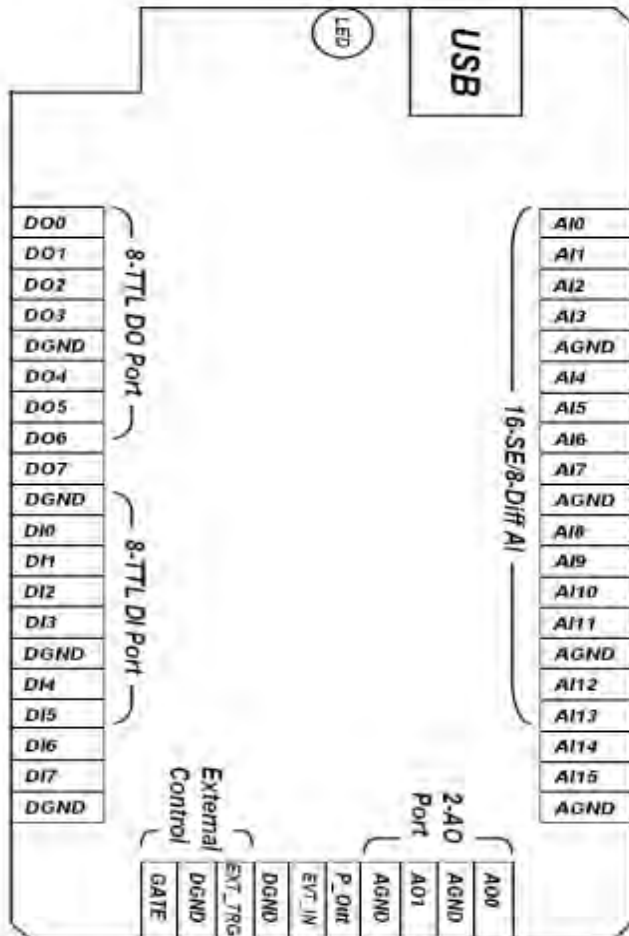
Pada tugas akhir ini untuk menghubungkan *processing unit* dengan *hardware* yang ada menggunakan *portable data acquisition module*. Untuk pembacaan *analog input*, *digital output*, dan *analog output* diperintah oleh *processing unit* via modul ini

3.2 Hardware

3.2.1 Advantech USB-4711A

Pada tugas akhir ini digunakan Advantech USB-4711A yang memiliki 8 port *digital input*, 8 port *digital output*, 2 port *analog output*,

16 port *analog input*, 4 port *digital ground*, 6 port *analog ground*, dan 3 port *external control*. Berikut konfigurasi port pada *Advantech USB-4711A* dan gambaran modulnya.



Gambar 3.11 Konfigurasi port *Advantech USB-4711A*



Gambar 3.12 Advantech USB-4711A

Pada tugas akhir ini digunakan 2 port *analog output* untuk pengatur *duty cycle* pada rangkaian pembangkit PWM (*Pulse Width Modulation*), 5 port *digital out* dimana 4 port untuk mengatur arah gerak motor (mengatur gerak vertikal dan horizontal) dan 1 port untuk pengatur penembakan, dan 1 port *analog input* untuk mengatur *emergency stop input* dari sensor *proximity*. Untuk lengkapnya penggunaan port pada Advantech USB-4711A dapat dilihat pada table berikut:

Tabel 3.1 Fungsi Penggunaan Port Advantech USB-4711

Port	Fungsi
DO0	Pin 1 Driver Motor Vertikal
DO1	Pin 2 Driver Motor Vertikal
DO2	Pin 1 Driver Motor Horizontal
DO3	Pin 2 Driver Motor Horizontal
DO4	Perintah menembak
AO0	Pengatur masukan Pembangkit PWM (Vertikal)
AO1	Pengatur masukan Pembangkit PWM (Horizontal)
AI0	Masukan Sensor <i>Proximity</i>
DGND	Referensi Ground
AGND	Referensi Ground

3.2.2 Sensor *Proximity*

Sensor *Proximity* disini digunakan untuk memberi batasan gerak pada gerakan horizontal (*emergency stop* pada sistem). Sensor *proximity*

yang digunakan adalah Autonics PR8-2DN. Sensor ini memiliki parameter sebagai berikut:

- Jarak deteksi : $2\text{mm} \pm 10\%$
- *Power Supply* : 12-24V
- Jenis Deteksi : Logam
- Output sensor : *Active Low*

Berikut gambaran sensor *Proximity* yang digunakan

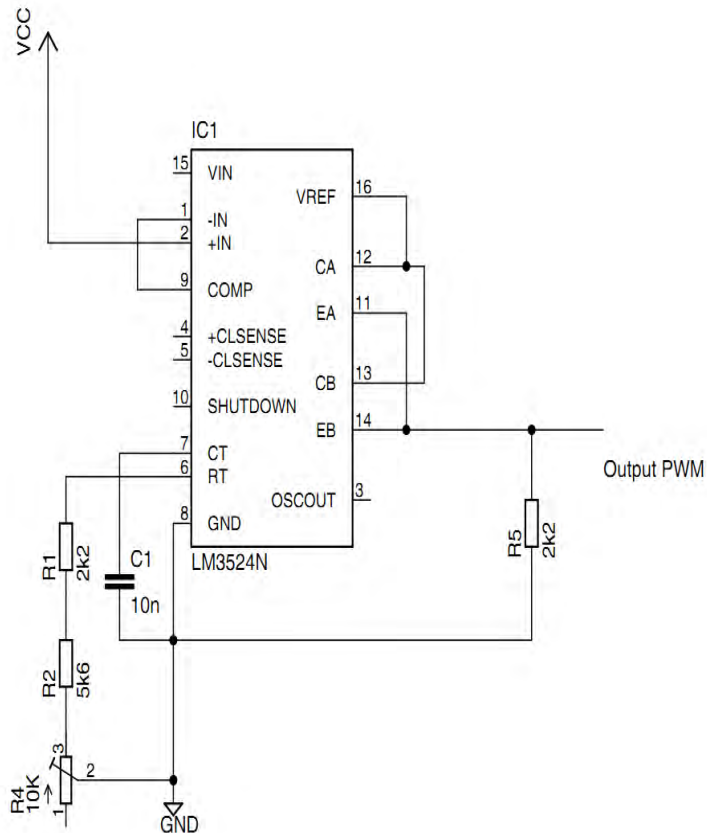


Gambar 3.13 Sensor *Proximity*

3.2.3 Rangkaian Pembangkit PWM (*Pulse Width Modulation*)

Pada blok ini menjelaskan rangkaian yang berfungsi untuk menghasilkan *pulse width modulation* yang akan masuk ke *driver* motor. Pada rangkaian ini dihasilkan *pulse width modulation* dari rentang *duty cycle* 0% hingga 100%. Rangkaian ini menggunakan IC LM3524 sebagai penghasil *pulse width modulation*. Referensi yang digunakan untuk mengatur *duty cycle* adalah masukan *analog input* dari Advantech

USB-4711A. Masukan untuk mengatur *duty cycle* adalah tegangan 0 volt hingga tegangan 5 volt. Berikut skematik rangkaian pembangkit *pulse width modulation*.



Gambar 3.14 Skematik rangkaian pembangkit PWM

Untuk menggerakkan motor horizontal dan vertikal diperlukan masing-masing 1 rangkaian, sehingga terdapat 2 buah rangkaian serupa pada rangkaian yang dibuat. Berikut gambaran dari rangkaian yang telah dibuat:



Gambar 3.15 Rangkaian pembangkit PWM (dalam kotak)

Prinsip kerja dari rangkaian tersebut adalah, semakin besar tegangan masukan pada port 2, maka semakin besar *duty cycle* yang dihasilkan pada port 7 dari IC 3524 tersebut. Tegangan keluaran yang diinginkan diambil pada port 11 dan 14. Untuk mengatur tegangan keluaran pada *duty cycle*, pada port 12 dan 13 diberikan tegangan masukan. Keluaran tegangan *duty cycle* mengikuti nilai tegangan yang memasuki port 12 dan 13 tersebut. Sedangkan untuk mengatur frekuensi keluaran pada keluaran, bergantung pada perbandingan resistansi dari resistor pada port 6 dan kapasitor pada port 7. Kedua rangkaian ini akan menghasilkan gelombang gerigi segitiga yang akan dikomparasikan dengan tegangan input dari port 2. Keluaran dari komparator tersebut mengatur waktu *on* dan *off* pada keluaran port 12 dan 13 sehingga keluaran akhir yang didapatkan adalah gelombang kotak dengan periode *on* (tegangan keluaran sama dengan tegangan masukan port 12 dan 13) dan *off* (tegangan keluaran 0volt) yang diinginkan. Perbandingan antara waktu *on* dan *off* itulah yang disebut dengan *duty cycle*. Berikut rumusan dari nilai *duty cycle*:

$$Duty Cycle = \frac{T_{on}}{T_{off} + T_{on}} \times 100\% \quad (3-7)$$

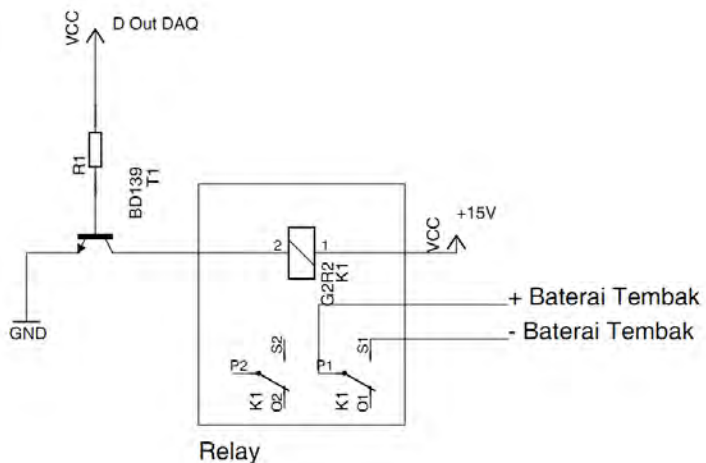
T_{on} : Waktu tegangan keluaran maksimal

T_{off} : Waktu tegangan keluaran 0 volt.

Keluaran dari rangkaian pembangkit PWM ini akan menuju ke *driver* motor untuk mengatur kecepatan motor. Keluaran rangkaian PWM vertikal akan menuju ke pin masukan PWM pada *driver* motor vertikal sedangkan keluaran rangkaian PWM horizontal akan menuju ke pin masukan PWM pada *driver* motor horizontal

3.2.4 Aktuator Tembak

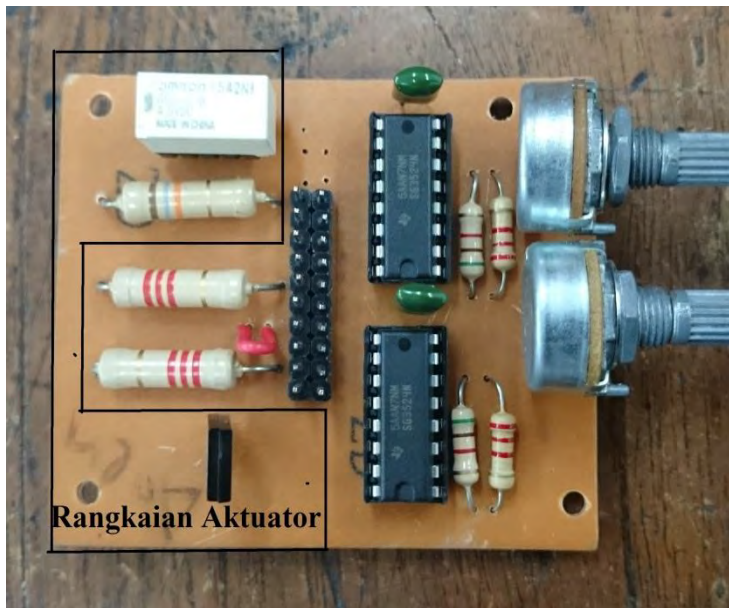
Disini dibuat sebuah rangkian untuk mengatur penembakan dari senjata yang digunakan. Berikut skematik dari rangkaian aktuator tembak yang telah dibuat:



Gambar 3.16 Skematik rangkaian aktuator tembak

Fungsi dari transistor BD139 pada rangkaian tersebut adalah untuk mengaktifkan relay. Saat basis pada transistor diberi tegangan, maka transistor akan mengalami saturasi dan membuat relay mengaktifkan

relay. Jika relay aktif, maka rangkaian yang sebelumnya *normally open* akan menjadi *close* sehingga rangkaian baterai tembak menjadi *close loop* dan membuat senjata yang dipasang menembak. Untuk menghentikan penembakan, maka cukup menghentikan tegangan basis menjadi 0. Jika tegangan basis 0volt, maka transistor BD139 menjadi *open circuit* pada emitter dan kolektornya sehingga membuat relay menjadi *off* dan rangkaian loop baterai senjata menjadi *open loop* dan menghentikan siklus menembak. Berikut rangkaian untuk aktuator menembak.



Gambar 3.17 Rangkaian aktuator menembak

3.2.5 *Driver Motor*

Driver Motor disini menggunakan BTN7970. Dalam tugas akhir ini menggunakan 2 buah *driver motor*. 1 untuk motor penggerak vertikal dan 1 lagi untuk motor penggerak horizontal. Berikut gambaran dari *driver motor* yang digunakan:



Gambar 3.18 Driver motor BTN7870B

Supply yang digunakan untuk *driver* motor menggunakan 24VDC 5A. Sedangkan masukan driver motor terdapat pin 3 yaitu masukan PWM untuk mengatur kecepatan motor, dan in 1 dan in 2 untuk mengatur arah gerakan motor. Untuk mengarahkan pergerakan pergerakan atas dan bawah pada motor penggerak vertikal dan pergerakan kiri dan kanan pada motor penggerak horizontal memiliki kondisi tertentu. Kondisi yang dimaksud dapat dilihat pada table berikut ini:

Tabel 3.2 Tabel kebenaran driver motor Vertikal

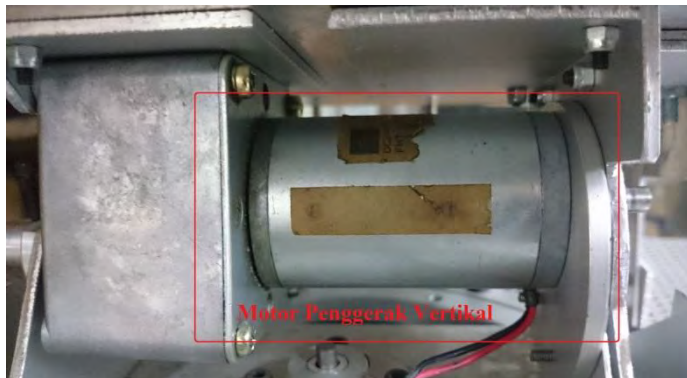
Driver Motor Vertikal		
In 1	In 2	Aksi pergerakan
0	0	Diam
0	1	Atas
1	0	Bawah
1	1	Diam

Tabel 3.3 Tabel kebenaran driver motor Horizontal

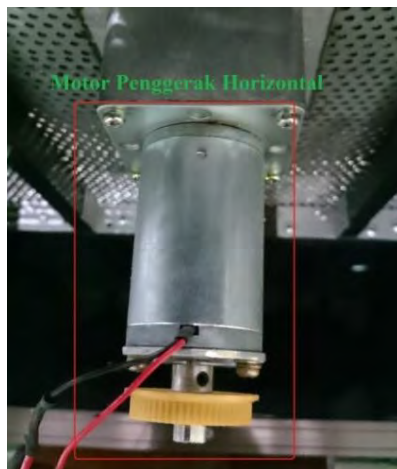
Driver Motor Horizontal		
In 1	In 2	Aksi pergerakan
0	0	Diam
0	1	Kanan
1	0	Kiri
1	1	Diam

3.2.6 Motor DC

Motor yang digunakan sistem penggerak horizontal dan vertikal pada tugas akhir ini adalah motor DC yang bisa bergerak pada 2 arah yang berlawanan. Arah pergerakan motor ditentukan oleh pemberian tegangan DC pada kedua input motor. Berikut gambaran motor DC yang digunakan pada tugas akhir ini beserta peletakannya.



Gambar 3.19 Motor DC penggerak vertikal



Gambar 3.20 Motor DC penggerak horizontal

3.2.7 Senjata

Senjata yang digunakan pada tugas akhir ini adalah jenis elektrik *air soft gun*. Terdapat 2 buah senjata yang digunakan yaitu purwarupa M4 dengan perbandingan ukuran 1:1 dan perbandingan berat 1:1 dan purwarupa SIG552 dengan ukuran 1:2 dan perbandingan berat 1:15. Berikut gambaran dari senjata yang digunakan:



Gambar 3.21 Purwarupa M4 1:1



Gambar 3.22 Purwarupa SIG552 1:2

3.2.8 Kamera

Kamera yang digunakan pada tugas akhir ini adalah Logitech C920. Berikut gambaran kamera yang digunakan:



Gambar 3.23 Logitech C920

Berikut spesifikasi dari Logitech C920:

- Perekaman video hingga 1080p (1920 x 1080 piksel)
- Kompresi video H.264
- Lensa Carl Zeiss dengan 20 langkah *autofocus*
- Built-in dual stereo mics* dengan koreksi noise otomatis
- Koreksi otomatis pada kondisi cahaya yang rendah
- Hi-Speed USB 2.0 (USB 3.0 ready)*
- Tripod ready clip*
- 15 *megapixels* untuk *capture* foto.

3.2.9 Komputer

Pada tugas akhir ini digunakan sebuah komputer sebagai *processing unit*. Spesifikasi komputer yang digunakan pada tugas akhir ini adalah:

- *Manufacture* : Dell
- *Model* : XPS L502X
- *CPU* : Intel ® Core™ i7-2630QM
@2.00GHz
- *RAM* : 4GB
- *System Type* : 64-bit

3.2.10 Sistem Penembak

Dalam bab ini dijelaskan perancangan keseluruhan sistem penembak. Berikut gambaran dari sistem penembak dan penjelasan penempatan *hardware* yang digunakan pada sistem meliputi *driver motor*, rangkaian pembangkit PWM, senjata, sensor *proximity*, kamera, *DAQ module* dan motor DC untuk kedua jenis senjata.



Gambar 3.24 Rancangan sistem penembak dengan purwarupa SIG552



Gambar 3.25 Rancangan sistem penembak dengan purwarupa M4

Berikut penjelasan penempatan hardware:

No:

- 1 = Senjata
- 2 = Kamera
- 3 = Motor DC penggerak vertikal
- 4 = Sensor *Proximity*
- 5 = Rangkaian pembangkit PWM
- 6 = Driver motor DC
- 7 = Rangkaian aktuator penembak
- 8 = Advantech USB-4711A

Untuk peletakan motor penggerak horizontal ada di dalam bodi sistem. Berikut gambaran posisi motor DC penggerak horizontal:



Gambar 3.26 Posisi motor DC penggerak horizontal

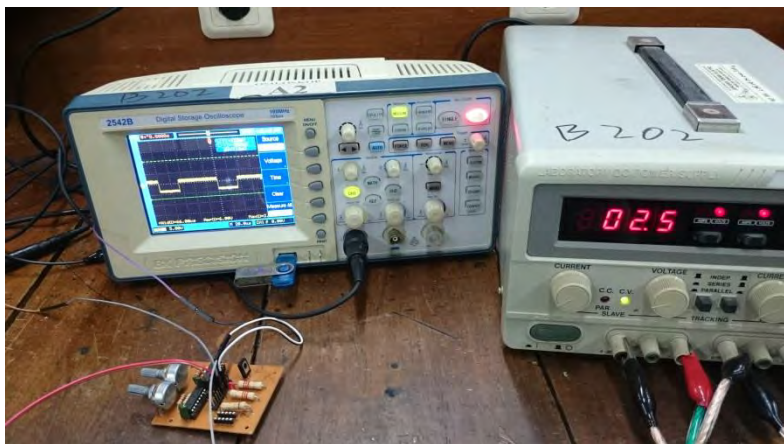
BAB IV

PENGUJIAN

Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang. Bab ini bertujuan untuk mengetahui apakah tujuan dalam perancangan sistem pada tugas akhir ini telah terlaksana atau tidak. Pengujian pada bab ini terdiri dari pengujian keluaran rangkaian pembangkit PWM, serta pelacakan dan penguncian terhadap sasaran dengan dua senjata yang berbeda.

4.1 Pengujian Rangkaian Pembangkit PWM

Pengujian yang dilakukan pada bagian ini adalah pengujian untuk mengambil atau menyimpan data gelombang PWM yang dihasilkan oleh rangkaian pembangkit PWM. Pengujian dilakukan dengan memberi tegangan referensi antara 0 hingga 4 Volt dimana akan dikonversikan menuju presentase *duty cycle*. Semakin besar nilai tegangan referensi yang diberikan, maka *duty cycle* keluaran dari rangkaian akan semakin besar. Berikut gambaran dari pengujian yang dilakukan:

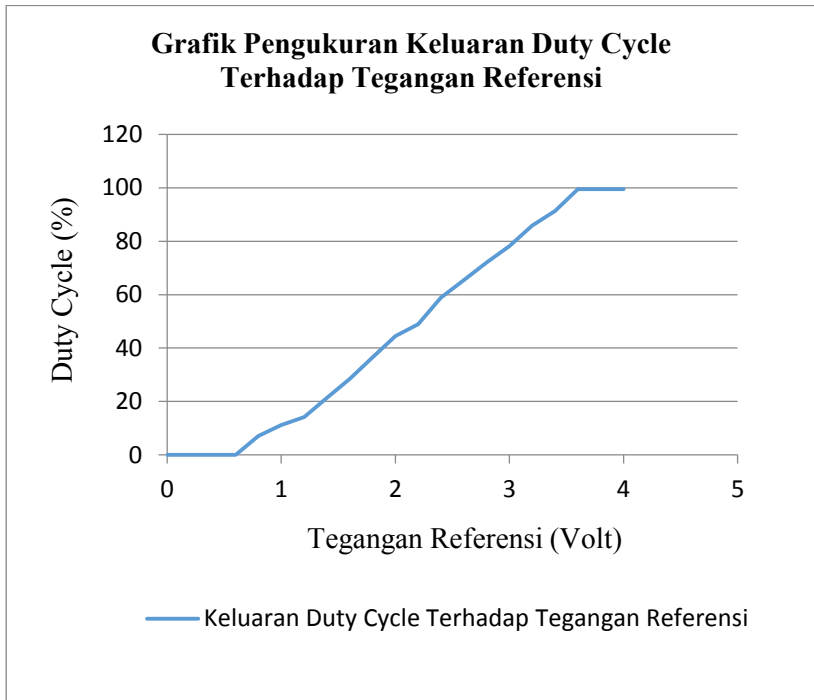


Gambar 4.1 Pengujian rangkaian pembangkit PWM

Pada pengujian ini dilakukan pengambilan data berupa keluaran *duty cycle* dari rangkaian pembangkit PWM (*Pulse Width Modulation*) tersebut setiap kenaikan 0,2volt pada tegangan referensi. Dari pengujian tersebut dapat diambil data-data sebagai berikut:

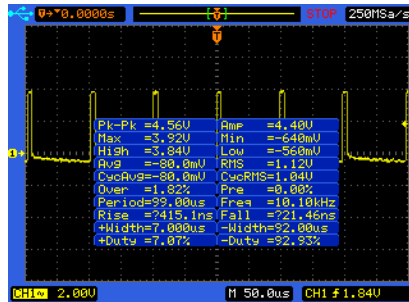
Tabel 4.1 Pengukuran keluaran *Duty Cycle* terhadap tegangan referensi

Tegangan referensi (Volt)	Duty Cycle (%)
0	0
0,2	0
0,4	0
0,6	0
0,8	7,07
1	11,17
1,2	14,14
1,4	21,32
1,6	28,43
1,8	36,45
2	44,44
2,2	48,98
2,4	58,88
2,6	65,48
2,8	72,08
3	78,17
3,2	85,79
3,4	91,37
3,6	99,49
3,8	99,49
4	99,49

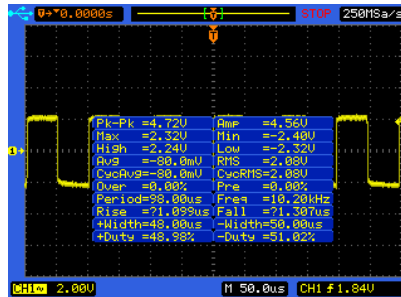


Gambar 4.2 Grafik hasil pengukuran keluaran *duty cycle* terhadap tegangan referensi

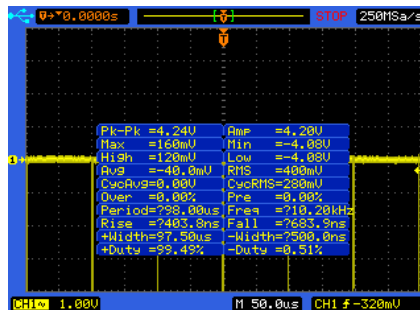
Dari data pengujian tersebut dapat diketahui bahwa keluaran *duty cycle* pada rangkaian pembangkit PWM yang telah dibuat dimulai pada saat tegangan referensi bernilai kurang dari 0,8 volt dan diatas 0,6volt serta mencapai nilai maksimum pada saat tegangan referensi bernilai 3,6volt. Dari data tersebut juga dapat diketahui bahwa setiap kenaikan tegangan sebanyak 0,1 volt terjadi kenaikan nilai *duty cycle* sebesar kurang lebih 3%. Untuk besaran tegangan *peak* keluaran pada rangkaian tersebut antara 0 hingga 4,56 volt Berikut beberapa gambar hasil pengukuran pada osiloskop:



Gambar 4.3 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 0,8Volt



Gambar 4.4 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 2,2Volt



Gambar 4.5 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 3,6Volt

4.2 Pengujian Motor

Seperti dijelaskan pada bab sebelumnya pada tugas akhir ini digunakan 2 jenis purwarupa senjata dengan berat yang berbeda. Pengujian ini dilakukan untuk mengetahui nilai minimal *duty cycle* yang dibutuhkan untuk menggerakkan motor dengan beban yang berbeda. Untuk motor vertikal ditambahkan parameter berupa pengujian gerak dari sudut maksimum. Dari pengujian didapat data-data sebagai berikut:

Tabel 4.2 Pengujian motor DC vertikal

Kondisi beban	Massa Beban	Duty Cycle Minimum	
		Sudut 0	Sudut Maksimum
tanpa beban	0Kg	25%	25%
SIG 552	0,5Kg	32%	65%
M4	4Kg	32%	-

Dari pengujian dapat dianalisis yaitu pada saat sistem diberikan beban, maka dibutuhkan *duty cycle* minimum yang lebih besar untuk menggerakkan motor. Hal ini dapat dilihat dari saat kondisi tanpa beban dan dengan beban dimana dari pengujian terlihat kondisi tanpa beban membutuhkan *duty cycle* yang lebih kecil untuk menggerakkan motor dan kondisi dengan beban membutuhkan *duty cycle* yang lebih besar untuk menggerakkan motor.

Sedangkan sudut memiliki pengaruh besar untuk menggerakkan motor. Saat sudut maksimum (saat senjata menunduk atau mendongak), dibutuhkan *duty cycle* yang lebih besar untuk menggerakkan motor dapat dilihat dari kondisi tanpa beban dan dengan beban SIG 552 dimana membutuhkan *duty cycle* yang lebih besar untuk menggerakkan motor daripada pada saat sudut 0. Untuk purwarupa M4 yang memiliki massa 4kg, beban yang diberikan melebihi kapasitas motor sehingga pada sudut maksimum diberikan *duty cycle* maksimal tetap tidak bergerak. Sudut maksimum untuk purwarupa SIG552 adalah 15° dan untuk 7° untuk purwarupa M4.

Tabel 4.3 Pengujian motor DC horizontal

Kondisi Beban	Masssa Senjata	Duty Cycle Minimum
Tanpa Beban	0Kg	30%
SIG552	0,5Kg	35%
M4	4Kg	40%

Dari pengujian dapat dianalisis yaitu saat diberikan beban, maka dibutuhkan *duty cycle* minimum yang lebih besar untuk menggerakkan motor. Hal ini dapat dilihat dari saat kondisi tanpa beban dan dengan beban dimana dari pengujian terlihat kondisi tanpa beban membutuhkan *duty cycle* yang lebih kecil untuk menggerakkan motor.

Untuk motor horizontal terjadi kondisi dimana motor terdapat hambatan untuk bergerak sehingga terjadi keterlambatan bergerak. Hal ini disebabkan oleh mekanik pada motor horizontal.

4.3 Pengujian Pendeteksi Manusia

Dengan metode yang telah dijelaskan sebelumnya, dilakukan pengujian apakah sistem dapat mendeteksi orang pada hasil citra yang ditangkap. Pengujian dilakukan dengan metode *rough search* dimana sistem akan mendeteksi setiap orang yang dideteksi. Pengujian dilakukan pada siang hari dengan pencahayaan matahari dan tidak berawan

- Waktu : Pukul 15.30 WIB
Tempat : Lapangan Parkir Dosen Teknik Elektro ITS
Ketinggian : 90cm

Pengujian dilakukan dengan jumlah orang yang dideteksi bertambah. Dimulai dengan 1 orang, 2 orang, dan terakhir 3 orang. Manusia yang dideteksi dalam kondisi menghadap atau membelakangi

kamera yang digunakan pada sistem. Dari pengujian tersebut didapat data-data sebagai berikut

Tabel 4.4 Tabel pengujian deteksi manusia

Jumlah Orang	Jumlah Terdeteksi
1	1
2	1
3	2



Gambar 4.6 Pengujian pendeteksi manusia



Gambar 4.7 Pengujian deteksi manusia



Gambar 4.8 Pengujian deteksi manusia



Gambar 4.9 Pengujian deteksi manusia

Dari data diatas dapat disimpulkan bahwa untuk pendeteksian manusia pada pengujian ini terdapat kesalahan deteksi berupa *false positive* yaitu dimana benda bukan manusia terdeteksi sebagai manusia pada gambar 4.8 dan terdapat kesalahan berupa objek manusia tidak terdeteksi pada gambar 4.6, 4.7, dan 4.8. Dari data tersebut dapat dibuktikan juga arah menghadap objek mempengaruhi deteksi. Arah menghadap yang optimal adalah objek manusia menghadap kamera secara langsung atau membelakangi kamera jika ada kemiringan mempengaruhi pendeteksian.

4.4 Pengujian Pelacakan dan Penguncian Sasaran pada Siang Hari

Dengan menggunakan metode yang telah dijelaskan pada bagian perancangan sistem, maka dilakukan pengujian terhadap sistem yang telah dibuat. Pada pengujian kali ini dilakukan dengan purwarupa

purwarupa SIG 552 Pengujian awal dilakukan pada siang hari dengan cuaca berawan dan kondisi pencahayaan yang cukup.


- Waktu : Pukul 14.30 WIB
- Tempat : Lapangan Parkir Dosen Teknik Elektro ITS
- Ketinggian : 90cm

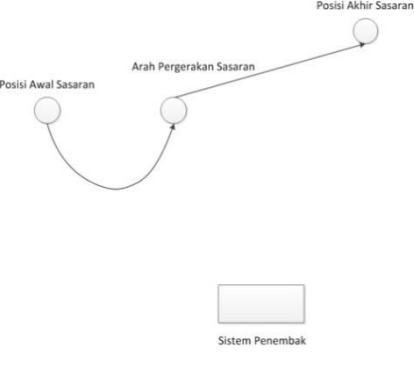

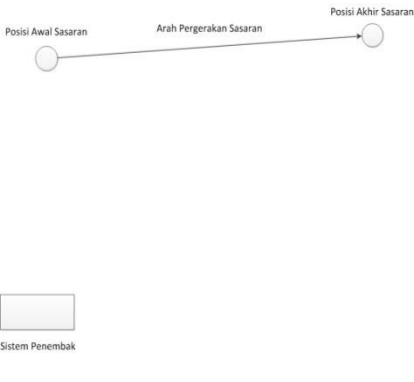
Pengujian dilakukan seberapa lama sistem dapat melacak dan mengunci sasaran dengan warna sasaran yang dirubah. Dari pengujian tersebut didapat data-data sebagai berikut:

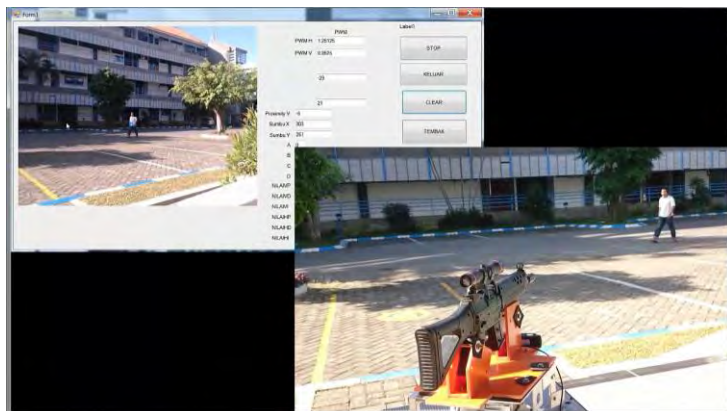
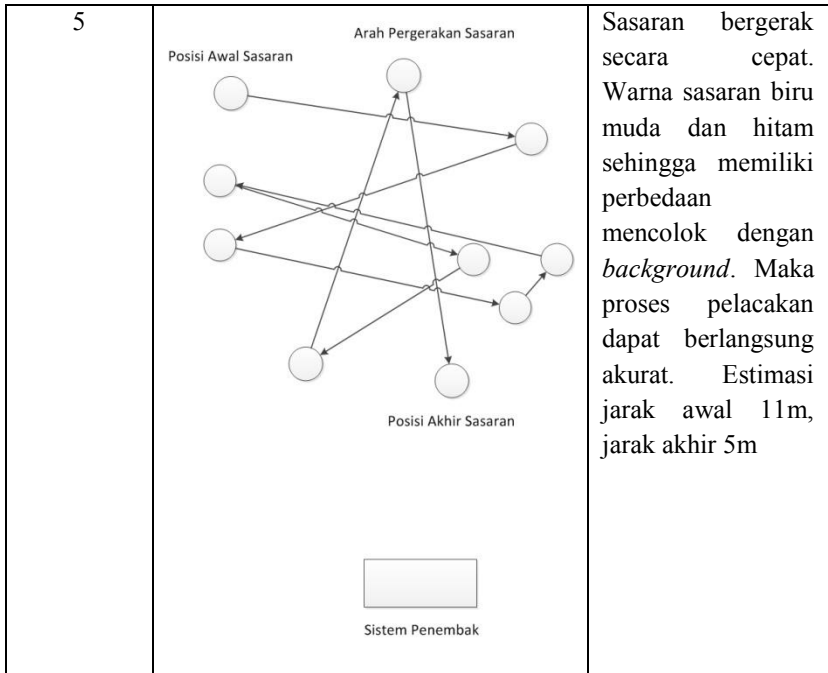
Tabel 4.5 Tabel pengujian pelacakan pada siang hari

Percobaan	Lama Pelacakan (detik)	Warna Sasaran
1	11	Biru
2	12	Biru
3	9	Merah
4	28	Merah
5	40	Biru Hitam

Tabel 4.6 Tabel alur gerak sasaran dan keterangan pengujian pelacakan pada siang hari

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna belakangnya sehingga sistem lepas dari sasaran. Estimasi jarak 7m akhir, 10m awal.</p>

2		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna belakangnya sehingga sistem lepas dari sasaran. Estimasi jarak awal 7m dan akhir 8m</p>
3		<p>Sasaran berwarna merah sehingga berbeda dengan warna <i>background</i>. Sasaran bergerak cepat sehingga sasaran terlepas dari sistem. Estimasi jarak 7m awal, 9m akhir.</p>
4		<p>Sasaran bergerak secara lambat. Warna sasaran merah sehingga berbeda dengan <i>background</i>. Maka proses pelacakan dapat berlangsung akurat. Estimasi jarak awal 7m, akhir 12m</p>



Gambar 4.10 Proses pengujian pelacakan pada siang hari

Selain itu dilakukan pengujian pelacakan kedua pada hari yang berbeda namun dengan posisi dudukan sistem senjata yang sama. Pengujian dilakukan pada kondisi cuaca cerah dan pencahayaan yang cukup. Pengujian dilakukan pada:

- Waktu : Pukul 13.30 WIB
Tempat : Lapangan Parkir Dosen Teknik Elektro ITS
Ketinggian : 90cm

Pengujian dilakukan seberapa lama sistem dapat melacak dan mengunci sasaran dengan 2 buah purwarupa senjata berbeda yang telah ada sebelumnya. Pada pengujian kali ini dilakukan dengan 2 jenis purwarupa yang berbeda yaitu purwarupa SIG 552 dengan massa purwarupa 0,5kg dan purwarupa M4 dengan massa purwarupa 4kg. Pada pengujian ini sasaran yang digunakan menggunakan warna yang tetap yaitu warna putih. Dari pengujian tersebut didapat data-data sebagai berikut:

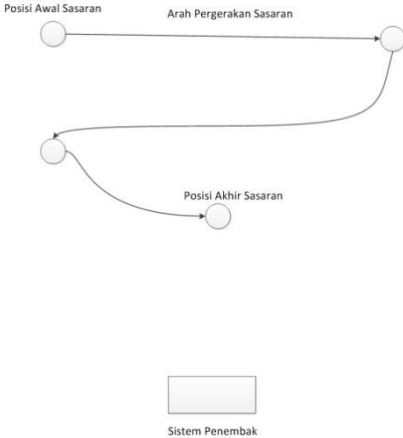
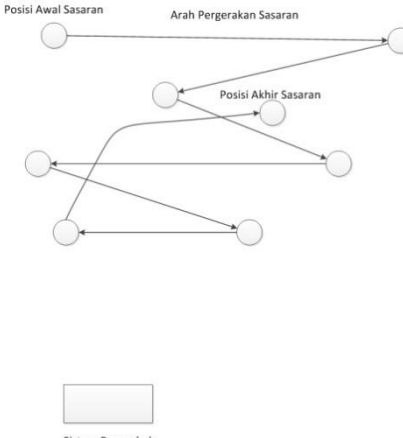
Tabel 4.7 Tabel pengujian pelacakan kedua pada siang hari purwarupa SIG552

Percobaan	Lama Pelacakan (detik)
1	42
2	61

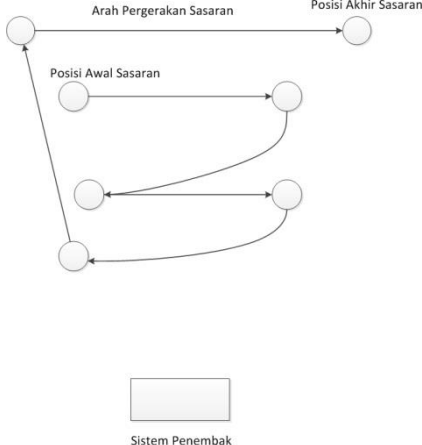
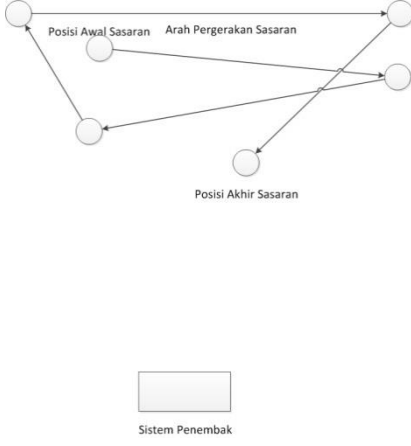
Tabel 4.8 Tabel pengujian pelacakan kedua pada siang hari purwarupa M4

Percobaan	Lama Pelacakan (detik)
1	43
2	48

Tabel 4.9 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada siang hari purwarupa SIG552

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Warna putih yang digunakan sasaran memiliki kontras warna yang berbeda dengan <i>background</i> yang gelap akibat terkena bayangan. Estimasi jarak awal 10m, jarak akhir 6m</p>
2		<p>Sasaran bergerak dengan kecepatan yang lebih cepat dari percobaan 1. Warna putih yang digunakan sasaran memiliki kontras warna yang berbeda dengan <i>background</i> yang gelap akibat terkena bayangan. Estimasi jarak awal 11m, jarak akhir 9m</p>

Tabel 4.10 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada siang hari purwarupa M4

Percobaan	Alur Pergerakan Sasaran	Keterangan
1	 <p>Posisi Awal Sasaran</p> <p>Arah Pergerakan Sasaran</p> <p>Posisi Akhir Sasaran</p> <p>Sistem Penembak</p>	<p>Sasaran bergerak secara lambat. Warna putih yang digunakan sasaran memiliki kontras warna yang berbeda dengan <i>background</i> yang gelap akibat terkena bayangan.</p>
2	 <p>Posisi Awal Sasaran</p> <p>Arah Pergerakan Sasaran</p> <p>Posisi Akhir Sasaran</p> <p>Sistem Penembak</p>	<p>Sasaran bergerak dengan kecepatan yang sama dengan percobaan sebelumnya. Warna putih yang digunakan sasaran memiliki kontras warna yang berbeda dengan <i>background</i> yang gelap akibat terkena bayangan.</p>

Dari data-data pada pengujian pelacakan pertama diatas dapat disimpulkan bahwa kecepatan gerak dan perbedaan warna sasaran dengan *background* memiliki pengaruh terhadap proses pelacakan. Jika sasaran bergerak dengan cepat maka sasaran mudah lepas dari sistem karena perbedaan piksel terhadap waktu pada *image* yang tertangkap terlalu besar sehingga pelacakan sasaran dengan metode *optical flow* mudah lepas. Sedangkan untuk perbedaan warna antara sasaran dengan *background* memiliki pengaruh yang besar juga. Karena metode ini melihat pergeseran *flow* dari warna pada piksel, jika terdapat gradasi warna yang rendah maka posisi sasaran bisa berubah dan dapat lepas dari sasaran.

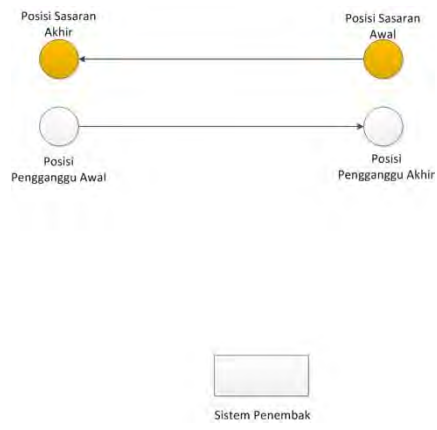
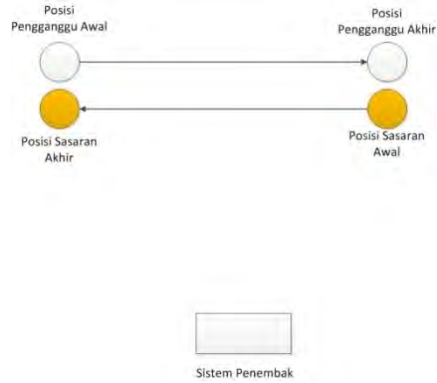
Pada pengujian pelacakan kedua, beban purwarupa yang digunakan memiliki pengaruh terhadap gerakan motor. Pada purwarupa SIG 552, pergerakan horizontal memiliki respon lebih cepat karena memerlukan tenaga gerak yang lebih kecil dari pada dengan beban purwarupa M4. Hal ini sesuai dengan pengujian motor pada pembahasan sebelumnya. Selain itu akibat dari beratnya beban M4, terjadi *overshoot* pada pergerakan motor dimana motor berhenti melebihi posisi yang ditentukan. Motor akan memperbaiki posisi kembali meskipun terjadi *overshoot*. Untuk pergerakan vertikal pada purwarupa M4 ditiadakan karena beratnya beban purwarupa menyebabkan pergerakan vertikal pada motor tidak memungkinkan.

Setelah dilakukan pengujian pelacakan, dilakukan pengujian lanjutan berupa gangguan pada sasaran yang telah dikunci dimana sasaran dilewati benda asing baik dibelakang atau didepan sasaran. Pengujian ini dilakukan pada kondisi cuaca cerah dan pencahayaan yang cukup di ruang terbuka.

- Waktu : Pukul 15.30 WIB
Tempat : Lapangan Parkir Dosen Teknik Elektro ITS
Ketinggian : 90cm

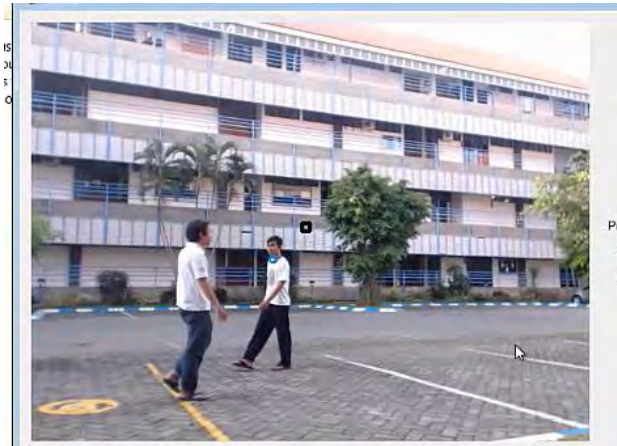
Pada pengujian ini diambil data apakah benda asing yang melewati sasaran yang telah dikunci baik melalui depan sasaran maupun belakang sasaran mempengaruhi penguncian sasaran apa tidak.

Tabel 4.11 Skenario pengujian gangguan sasaran

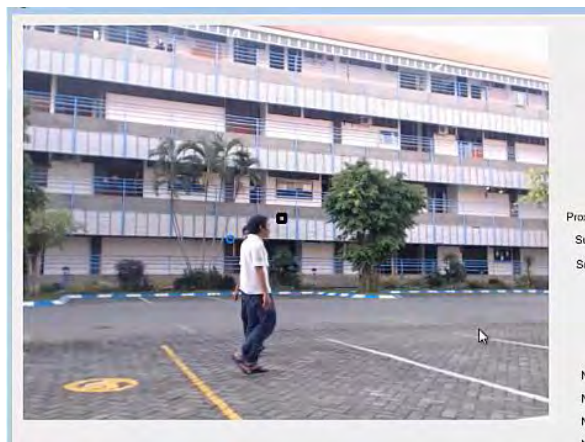
Percobaan	Alur Gangguan	Keterangan
1	 <p>Posisi Sasaran Akhir</p> <p>Posisi Sasaran Awal</p> <p>Posisi Pengganggu Awal</p> <p>Posisi Pengganggu Akhir</p> <p>Sistem Penembak</p>	<p>Sasaran diganggu dengan dilewati pengganggu di depannya sehingga sasaran sempat tertutup oleh pengganggu. Estimasi jarak sasaran 14m hingga 18m.</p>
2	 <p>Posisi Pengganggu Awal</p> <p>Posisi Sasaran Akhir</p> <p>Posisi Pengganggu Akhir</p> <p>Posisi Sasaran Awal</p> <p>Sistem Penembak</p>	<p>Sasaran diganggu dengan dilewati pengganggu di depannya sehingga pengganggu sempat tertutup oleh sasaran. Estimasi jarak sasaran 5m hingga 7m.</p>

Dari data yang didapat pada pengujian, didapat bahwa sasaran yang diganggu oleh pengganggu di depan sasaran mengalami gangguan pelacakan dan penguncian sasaran. Hal ini terjadi akibat sasaran tertutup oleh pengganggu yang melewatinya. Pada pengujian kedua dimana sasaran berada didepan pengganggu yang melewatinya, sistem tetap

mengunci sasaran tanpa adanya gangguan karena pengganggu yang melewati sasaran tidak menutupi sasaran.



Gambar 4.11 Pengujian gangguan di depan sasaran sebelum gangguan melewati sasaran



Gambar 4.12 Pengujian gangguan di depan sasaran sesudah gangguan menutupi sasaran

4.5 Pengujian Pelacakan dan Penguncian Sasaran di Malam Hari

Pada bagian ini dilakukan pengujian pada kondisi cahaya yang kurang terhadap sistem yang telah dibuat. Pada pengujian kali ini dilakukan dengan satu buah purwarupa senjata yaitu purwarupa SIG 552. Pengujian dilakukan pada malam hari dengan penerangan lampu. Pengujian dilakukan pada:

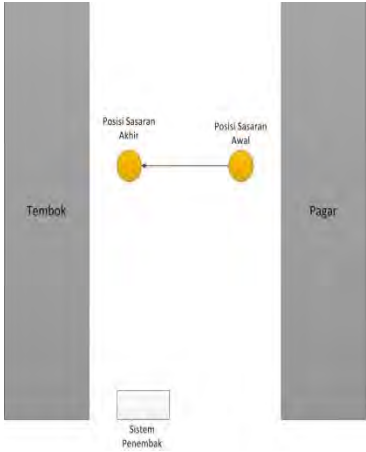
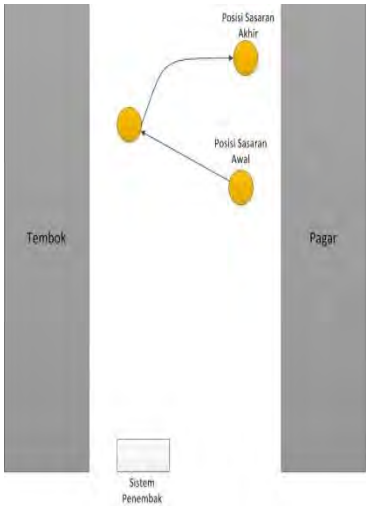
- Waktu : Pukul 19.00 WIB
Tempat : Koridor Lantai 2 Gedung B Jurusan Teknik Elektro ITS
Ketinggian : 90cm

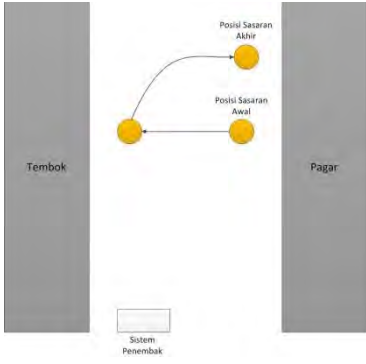
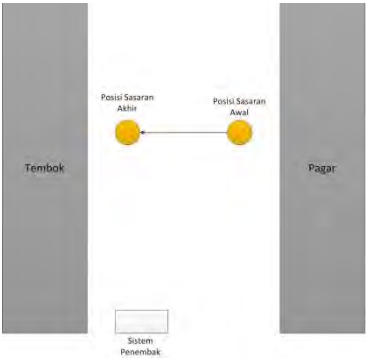
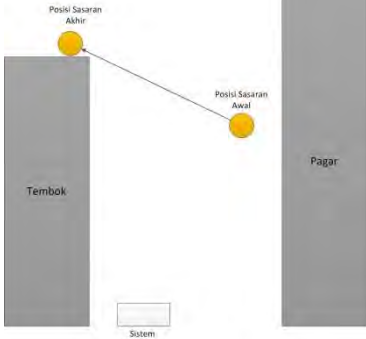
Pada pengujian ini dilakukan untuk mengetahui seberapa lama sistem dapat melacak dan mengunci sasaran dengan warna sasaran yang berbeda-beda namun dengan kondisi *background* dan posisi sistem senjata yang sama dan tidak dirubah. Selain itu pergerakan sasaran pada pengujian kali ini diatur memiliki jarak 4m hingga 5m saja. Kondisi pengujian dilakukan di lorong yang memiliki lebar area pergerakan sasaran sebesar 1,5m. Dari pengujian yang dilakukan tersebut didapat data-data sebagai berikut:

Tabel 4.12 Tabel pengujian pelacakan pada malam hari 1

Percobaan	Lama Pelacakan (detik)	warna sasaran
1	2	abu abu
2	5	abu abu
3	0	biru
4	2	merah
5	9	merah

Tabel 4.13 Tabel alur gerak sasaran dan keterangan pengujian pelacakan pada siang hari 1

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna belakangnya serta pencahayaan yang kurang, sehingga sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4m.</p>
2		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna belakangnya serta pencahayaan yang kurang, sehingga sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4-5m.</p>

3		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna pagar serta pencahayaan yang kurang, sehingga sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4-5m.</p>
4		<p>Sasaran bergerak secara lambat. Warna sasaran memiliki kontras warna yang berbeda dengan warna belakangnya namun pencahayaan yang kurang menyebabkan sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4m.</p>
5		<p>Sasaran bergerak secara lambat. Warna sasaran memiliki kontras warna yang berbeda dengan warna belakangnya. Sasaran terlepas dari pelacakan karena tertutup oleh tembok. Estimasi jarak sasaran 4-5m.</p>

Selain itu dilakukan pengujian pelacakan kedua pada hari yang sama dengan kondisi pengujian dilakukan pada malam hari dengan penerangan lampu seperti pada kondisi pengujian malam sebelumnya. Namun dengan jenis senjata yang berbeda dan sasaran yang sama. Pengujian dilakukan pada:

- Waktu : Pukul 20.00 WIB
Tempat : Koridor Lantai 2 Gedung B Teknik Elektro ITS
Ketinggian : 90cm

Pengujian dilakukan seberapa lama sistem dapat melacak dan mengunci sasaran dengan sasaran yang sama namun dengan jenis purwarupa senjata yang berbeda. Pada pengujian kali ini dilakukan dengan 2 jenis purwarupa yang berbeda yaitu purwarupa SIG 552 dengan massa 0,5kg dan purwarupa M4 dengan massa 4kg. Pada pengujian ini sasaran yang digunakan menggunakan warna yang tetap yaitu warna merah. Dari pengujian tersebut didapat data-data sebagai berikut:

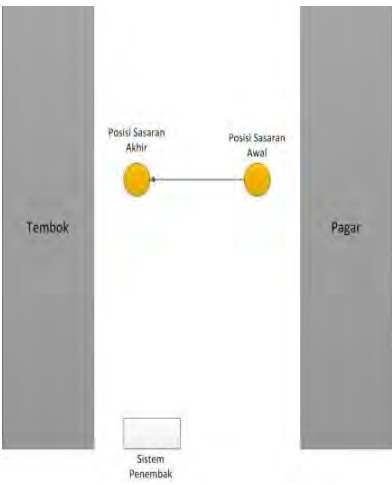
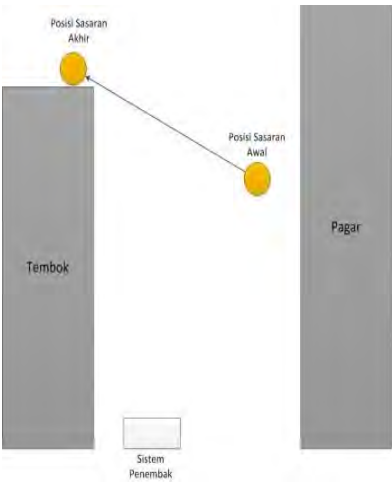
Tabel 4.14 Tabel pengujian pelacakan kedua pada malam hari purwarupa SIG552

Percobaan	Lama Pelacakan (detik)
1	2
2	9

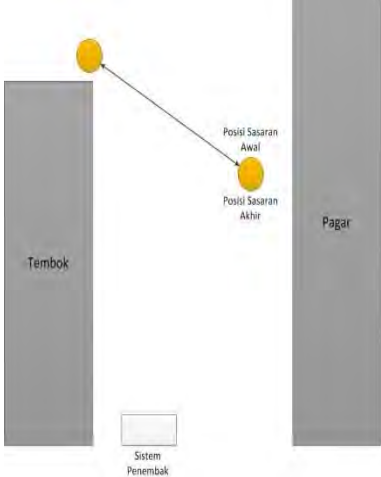
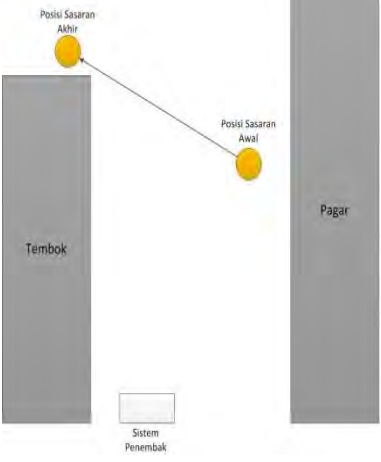
Tabel 4.15 T abel pengujian pelacakan kedua pada malam hari purwarupa M4

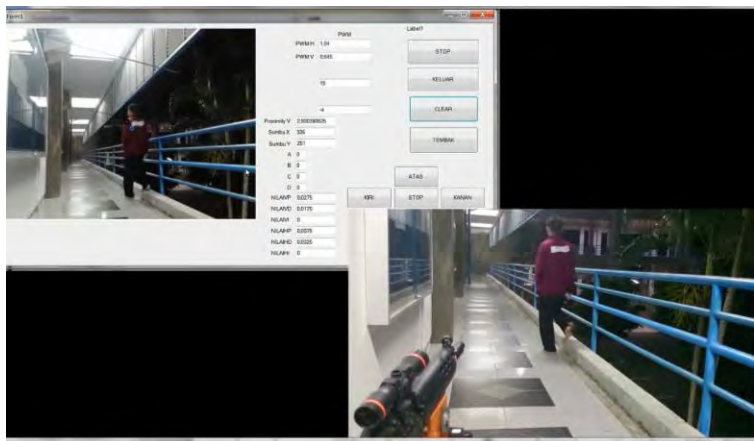
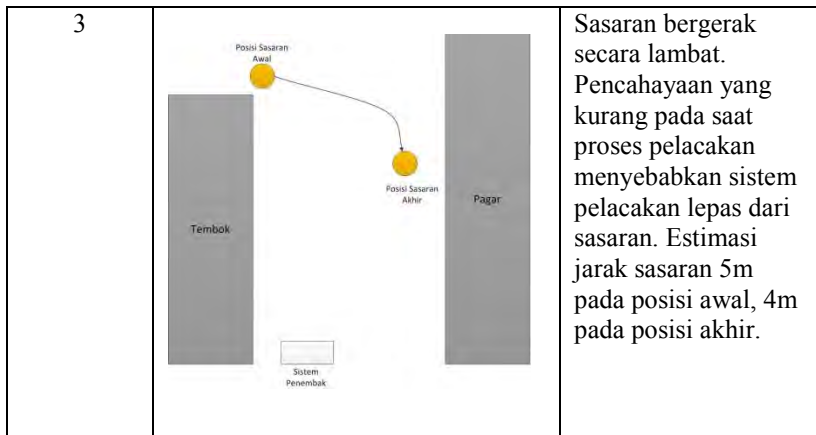
Percobaan	Lama Pelacakan (detik)
1	20
2	15
3	15

Tabel 4.16 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada malam hari purwarupa SIG552

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Pencerayaan yang kurang pada saat proses pelacakan menyebabkan sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4m.</p>
2		<p>Sasaran bergerak secara lambat. Pencerayaan yang kurang pada saat proses pelacakan menyebabkan pelacakan mengalami gangguan. Sasaran terlepas dari pelacakan karena sasaran tertutup oleh tembok. Estimasi jarak sasaran 4-5m.</p>

Tabel 4.17 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada malam hari purwarupa M4

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Pencapaian yang kurang pada saat proses pelacakan menyebabkan sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 5m pada posisi awal, 4m pada posisi akhir.</p>
2		<p>Sasaran bergerak secara lambat. Pencapaian yang kurang pada saat proses pelacakan menyebabkan pelacakan mengalami gangguan. Sasaran terlepas dari pelacakan karena sasaran tertutup oleh tembok. Estimasi jarak sasaran 4-5m.</p>



Gambar 4.13 Proses pengujian pelacakan pada malam hari

Dari data-data pada pengujian pelacakan pada malam hari yang pertama diatas dapat disimpulkan bahwa kondisi cahaya dan perbedaan warna sasaran dengan *background* memiliki pengaruh terhadap proses pelacakan. Cahaya yang kurang terang menyebabkan proses pengambilan gambar lambat, (*frame per detik*) sehingga menyebabkan

gerak motor mengalami kelebihan dan pergeseran *image* mengalami *blur*. Sedangkan untuk perbedaan warna antara sasaran dengan *background* memiliki pengaruh yang besar juga. Karena metode ini melihat pergeseran *flow* dari warna pada piksel, jika terdapat gradasi warna yang rendah maka posisi sasaran bisa berubah dan dapat lepas dari sasaran.

Pada pengujian pelacakan kedua, seperti pada peengujian malam hari yang pertama, kondisi cahaya mempengaruhi proses pelacakan. Selain itu beban purwarupa yang digunakan memiliki pengaruh terhadap gerakan motor. Pada purwarupa SIG 552, pergerakan horizontal memiliki respon lebih cepat karena memerlukan tenaga gerak yang lebih kecil dari pada dengan beban purwarupa M4. Hal ini sesuai dengan pengujian motor pada pembahasan sebelumnya. Selain itu akibat dari beratnya beban M4, terjadi *overshoot* pada pergerakan motor dimana motor berhenti melebihi posisi yang ditentukan. Motor akan memperbaiki posisi kembali meskipun terjadi *overshoot*. Untuk pergerakan vertikal pada purwarupa M4 ditiadakan karena beratnya beban purwarupa menyebabkan pergerakan vertikal pada motor tidak memungkinkan.

4.6 Pengujian Penembakan

Dengan menggunakan metode yang telah dijelaskan pada bagian perancangan sistem, maka dilakukan pengujian penembakan terhadap sistem yang telah dibuat. Pengujian bertujuan untuk menganalisis keakuratan penguncian sasaran pada sistem. Pada pengujian kali ini dilakukan dengan 1 jenis purwarupa yaitu purwarupa SIG 552. Sasaran yang digunakan memiliki diameter 20cm. Pengujian ini dilakukan pada ruang tertutup dengan pencahayaan yang cukup.


- Waktu : Pukul 19.30 WIB
Tempat : Laboratorium B202 Jurusan Teknik Elektro
Ketinggian : 120cm



Pengujian dilakukan dengan ketinggian sasaran dan jarak sasaran yang berbeda-beda. Sebelum dilakukan penembakan, sistem terlebih dahulu mengunci sasaran. Dari pengujian tersebut didapat data-data sebagai berikut:




Tabel 4.18 Tabel pengujian tembak.

Percobaan	Jarak Sasaran	Ketinggian Sasaran
1	5m	150cm
2	4m	150cm
3	3m	150cm
4	4m	120cm
5	5m	120cm
6	3m	120cm

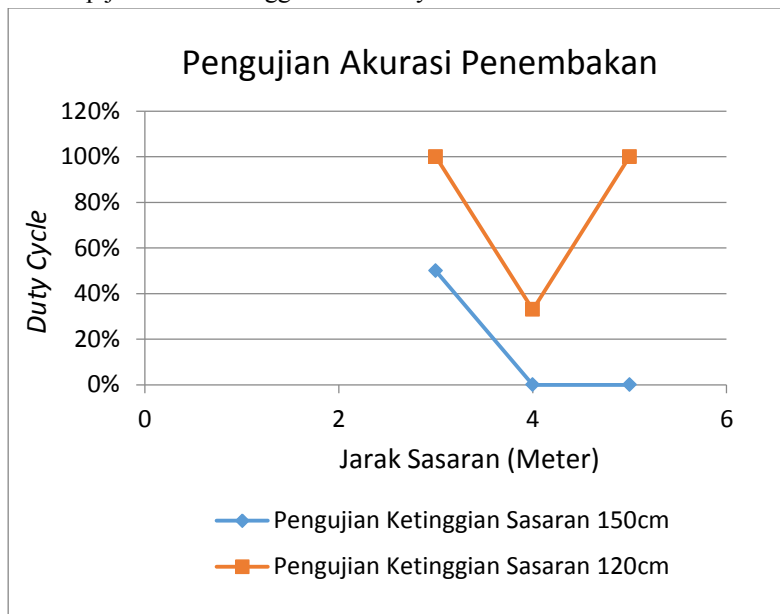
Tabel 4.19 Tabel hasil pengujian tembak

Percobaan	Peluru Ditembakkan	Peluru Kena Sasaran	Gambar
1	3	0	

2	3	0	
3	2	1	

4	3	1	
5	4	4	
6	4	4	

Dari data yang didapat dapat disimpulkan akurasi penembakan terhadap jarak dan ketinggian sasaran yaitu:



Gambar 4.14 Grafik Akurasi Penembakan Sasaran

Pada pengujian ini dianggap terkena sasaran jika peluru penembakan mengenai lingkaran dengan diameter 20cm tersebut (minimal lingkaran berwarna biru). Sasaran tersebut sesuai dengan standar dari ISSF (*International Shooting Sport Federation*) untuk sasaran berjarak 50 yard dengan senjata pistol.

Pada sistem dirancang dengan perhitungan gerakan GLBB dimana kecepatan least peluru dari sistem purwarupa SIG552 ini adalah kurang lebih 120kaki per detik dan dengan ketinggian sistem 120cm, jarak maksimum yang dapat ditempuh dengan ketinggian sasaran minimal 120cm adalah 20meter.

Dari data-data tersebut dapat dilihat bahwa ketinggian dan jarak sasaran mempengaruhi akurasi penembakan dimana saat sasaran memiliki ketinggian lebih dari ketinggian sistem, akurasi penembakan menurun. Selain itu semakin jauh jarak sasaran terhadap sistem juga mempengaruhi akurasi. Semakin jauh sasaran dari sistem penembak, akurasi penembakan semakin menurun.

Sistem penembakan ini adalah *openloop* sehingga kompensasi pada salah penembakan pada sistem tidak ada. Untuk diluar sistem, kesalahan penembakan dapat dituntut di pengadilan dengan sistem peradilan pidana. Di pengadilan tersebut akan ditentukan apakah ada suatu kompensasi ataupun hukuman akibat salah tembak atau tidak. [13]



Gambar 4.15 Proses pengujian penembakan

Halaman ini sengaja dikosongkan

BAB V

KESIMPULAN DAN SARAN

Setelah tahap pengujian sistem, dilakukan pengujian sistem untuk beberapa kondisi cahaya

Kesimpulan

Kesimpulan yang diperoleh dalam Tugas Akhir ini adalah :

1. Massa dari Senjata mempengaruhi respon sistem penggerak vertikal maupun horizontal. Semakin besar massa senjata, maka beban pada motor semakin besar dan menyebabkan respon motor melambat. Dengan massa purwarupa senjata yang diberikan adalah 0,5kg untuk SIG 552 dan 4kg untuk M4.
2. Performa sistem dengan metode *optical flow* untuk perhitungan pelacakan dengan *sampling time* setiap 40ms dan *frame* 480p tidak terjadi keterlambatan. *Sampling time* 40ms tersebut cukup untuk melakukan perhitungan dengan metode *optical flow* dengan spesifikasi komputer yang digunakan pada tugas akhir ini.
3. Hasil pengujian pelacakan pada sasaran didapat sasaran paling tidak memiliki gradasi warna yang berbeda dengan *background*. Sasaran memiliki perbedaan intensitas minimal 30 perubahan untuk kanal warna RGB 24 bit.
4. Hasil pengujian penembakan dengan standar sasaran ISSF untuk jarak 50yard memiliki rata-rata akurasi lebih dari 50% pada jarak sasaran tidak kurang dari 5m dengan sasaran memiliki ketinggian sejajar dengan sistem.

Saran

Beberapa saran yang penulis dapat berikan untuk pengembangan tugas akhir adalah sebagai berikut :

1. Sensor visual yang digunakan diharapkan lebih memiliki kualitas *Frame Per Second* (FPS) yang lebih baik dengan *frame per second* diatas 25FPS.

2. Diharapkan untuk menambah akurasi diberi perhitungan jenis peluru yang digunakan.
3. Demi mendapatkan proses yang lebih cepat dengan *Frame Per Second* (FPS) yang lebih baik, penggunaan *processing unit* dapat diganti dengan PC yang memiliki *processor* lebih cepat.

LAMPIRAN

Berikut adalah listing program dari tugas akhir saya,

```
Imports Emgu.CV
Imports Emgu.CV.CvEnum
Imports Emgu.CV.Util
Imports Emgu.CV.Structure
Imports Emgu.Util
Imports Emgu.CV.Features2D
Imports Emgu.CV.UI

Public Class Form1

    Dim Tengah As Rectangle

    Dim manualH As Double
    Dim manualL As Double
    Dim AO0 As New InstantAoCtrl
    Dim AO1 As New InstantAoCtrl
    Dim AI0 As New InstantAiCtrl
    Dim AI1 As New InstantAiCtrl
    Dim DigitalOut As New InstantDoCtrl
    Dim DataOut0 As Double
    Dim DataOut1 As Double
    Dim DataOutD As Double
    Dim NILAIHP As Double
    Dim NILAIVP As Double
    Dim NILAIVD As Double
    Dim NILAIHD As Double
    Dim NILAIHI As Double
    Dim NILAIVI As Double
    Dim DELTAERRORV As Double
    Dim DELTAERRORH As Double
    Dim PWMV As Double
    Dim PWMH As Double
    Dim XP As Double
    Dim YP As Double
    Dim YD As Double
    Dim XD As Double
    Dim XI As Double
```

```

Dim YI As Double
Dim Minus As Double
Dim DataIn1 As Double
Dim DataIn2 As Double

Dim tembak As Integer
Dim a As Integer
Dim b As Integer
Dim c As Integer
Dim d As Integer
Dim hasil As Integer
Dim errorx As Integer
Dim errory As Integer
Dim errorx1 As Integer
Dim errory1 As Integer
Dim errorxi As Integer
Dim erroryi As Integer
Dim titikengahx As Integer
Dim titikengahy As Integer

Dim xx As Integer
Dim yy As Integer

Dim capdev As New Capture(1)
Dim citraasli As New Image(Of Bgr, Byte)(640, 480)
Dim imColor As New Image(Of Bgr, Byte)(640, 480)
Dim imGray As New Image(Of Gray, Byte)(640, 480)
Dim imGray2 As New Image(Of Gray, Byte)(640, 480)
Dim bunder As Rectangle
Dim roi_wajah As Rectangle

Dim face As New
HaarCascade("haarcascade_upperbody.xml")
Dim faceDetected As MCvAvgComp()
Dim thershold As Double

Private captureInProgress As Boolean
Const radius = 5

```



```

Dim defineCorners As Boolean = False
Dim puntiA(0)() As PointF
Dim puntiB(0)() As PointF
Dim puntiTemp As PointF()()
Dim puntiBtemp(0)() As PointF
Dim add_remove_pt As Boolean = False
Dim count As Integer = 0
Dim lkcount, plk1, plk2, plk3, plk4, plk5, delplk As
Integer
Dim frame As Emgu.CV.Image(Of Emgu.CV.Structure.Bgr,
Byte)
Dim grayA As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim grayB As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim pyrBufferA As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim pyrBufferB As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim imgTemp As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim flags As Emgu.CV.CvEnum.LKFLOW_TYPE =
Emgu.CV.CvEnum.LKFLOW_TYPE.DEFAULT
Dim status As Byte()
Dim errors As Single()

Dim posisiX, posisiY, dposisiX, dposisiY, pposisiX,
pposisiY As Integer
Dim clickcount, klik_status, flag_lk As Integer

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
 MyBase.Load

capdev.SetCaptureProperty(CAP_PROP.CV_CAP_PROP_FRAME_WID
TH, 640)

capdev.SetCaptureProperty(CAP_PROP.CV_CAP_PROP_FRAME_HEI
GHT, 480)
    flag_lk = 0

```

```

manualH = 2
manualL = 2
AO0.SelectedDevice = New DeviceInformation(1)
AO1.SelectedDevice = New DeviceInformation(1)
DigitalOut.SelectedDevice = New
DeviceInformation(1)
AI0.SelectedDevice = New DeviceInformation(1)
AI1.SelectedDevice = New DeviceInformation(1)

a = 0
b = 0
c = 0
d = 0
titiktengahx = 320
titiktengahy = 240
tembak = 0
YI = 0
YD = 0
XD = 0
XI = 0

'Timer1.Enabled = True
End Sub

Private Sub Timer1_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
    citraasli = capdev.QueryFrame
    imColor = citraasli

    Tengah = New Rectangle(310, 230, 10, 10)
    citraasli.Draw(Tengah, New Bgr(Color.Black), 3)
    lukas()
    FaceDetection()
    PictureBox1.Image = citraasli.ToBitmap

    TextBox15.Text = dposisiX
    TextBox16.Text = dposisiY

```

```
TextBox17.Text = pposisiX  
TextBox18.Text = pposisiY
```

```
TextBox1.Text = errorx  
TextBox2.Text = errory  
TextBox3.Text = a  
TextBox4.Text = b  
TextBox5.Text = c  
TextBox6.Text = d  
TextBox7.Text = posisiX  
TextBox8.Text = posisiY
```

```
errorx = posisiX - 320  
errory = posisiY - 240
```

```
AI0.Read(0, DataIn1)  
AI1.Read(1, DataIn2)  
TextBox9.Text = DataIn1
```

```
    If DataIn2 > 2 Then  
'proximity Horizontal  
        a = a  
        b = b  
        c = 0  
        d = 0  
    End If
```

```
    If errorx < 32 Then  
'kanan dan kiri berhenti  
        a = a  
        b = b  
        c = 0  
        d = 0  
    End If
```

<pre> If errory < 24 Then bawah berhenti a = 0 b = 0 c = c d = d End If </pre>	<pre> 'atas </pre>
<pre> If errory < -24 Then a = 0 b = 1 c = c d = d End If </pre>	<pre> 'atas </pre>
<pre> If errory > 24 Then a = 1 b = 0 c = c d = d End If </pre>	<pre> 'bawah </pre>
<pre> If errorx < -32 Then a = a b = b c = 1 d = 0 End If </pre>	<pre> 'kanan </pre>
<pre> If errorx > 32 Then a = a b = b c = 0 d = 1 End If </pre>	<pre> 'kiri </pre>

```

        If errorx = -320 And errory = -240 Then
'berhenti total, over pixel

            a = 0
            b = 0
            c = 0
            d = 0
        End If

        If DataIn1 < 0.5 Then                                'proximity
Vertikal
            a = a
            b = b
            c = 0
            d = 0
        End If

        hasil = 8 * d + 4 * c + 2 * b + 1 * a
        DataOutD = 16 * tembak + hasil
        If hasil = 0 Then
'berhenti
            ' DataOutD = 0
            DigitalOut.Write(0, DataOutD)
        End If
        If hasil = 1 Then
'bawah
            'DataOutD = 1
            DigitalOut.Write(0, DataOutD)
        End If
        If hasil = 2 Then                                    'atas
            'DataOutD = 2
            DigitalOut.Write(0, DataOutD)
        End If
        If hasil = 4 Then                                    'kiri
            'DataOutD = 4

```

```

        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 8 Then
'kanan
        'DataOutD = 8
        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 5 Then
'kiri
'atas
        'DataOutD = 5
        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 6 Then
'kiri
'bawah
        'DataOutD = 6
        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 9 Then
'kanan atas
        ' DataOutD = 9
        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 10 Then
'kanan bawah
        ' DataOutD = 10
        DigitalOut.Write(0, DataOutD)
    End If

    'Fungsi PWM
    If errorx < 0 Then
        XP = (-1 * errorx)
    End If
    If errorx > 0 Then
        XP = errorx
    End If
    If errory < 0 Then
        YP = (-1 * errory)
    End If
    If errory > 0 Then
        YP = errory
    End If

```

```

End If

NILAIVP = ((YP / 240) * 0.6) 'Control
P Vertikal
If (YP / 240) > 0.9999 Then
    NILAIVP = 0
End If
DELTAERRORV = ((YP / 240) - (YD / 240))
If DELTAERRORV < 0 Then
    DELTAERRORV = (-1 * DELTAERRORV)
End If
NILAIVD = ((DELTAERRORV / 40) * 24) 'Control
D Vertikal
YD = YP

If errorry > -240 Then
    errorryi = errorry
End If
If errorry = -240 Then
    errorryi = 0
End If
YI = YI + (errorryi / 240)
'Control I Vertikal

If YP < 24 Then
    YI = 0
End If
NILAIVI = (YI / 40) * 1.4
If NILAIVI < 0 Then
    NILAIVI = NILAIVI * -1
End If

PWMV = NILAIVP + NILAIVD + NILAIVI + 0.6 '
'Control PID Vertikal
If PWMV > 3 Then
    PWMV = 3

```

```

End If

TextBox11.Text = PWMV
AO0.Write(0, PWMV)

NILAIHP = ((XP / 320) * 0.4) 'Control
P Horizontal
If (XP / 320) > 0.9999 Then
    NILAIHP = 0
End If
DELTAERRORH = ((XP / 320) - (XD / 320))
If DELTAERRORH < 0 Then
    DELTAERRORH = (-1 * DELTAERRORH)
End If
NILAIHD = ((DELTAERRORH / 40) * 32)
'Control D Horizontal
XD = XP

If errorx > -320 Then
    errorxi = errorx
End If
If errorx = -320 Then
    errorxi = 0
End If
XI = XI + (errorxi / 320)
'Control I Horizontal

If XP < 32 Then
    XI = 0
End If
NILAIHI = (XI / 40) * 2
If NILAIHI < 0 Then
    NILAIHI = NILAIHI * -1
End If

```



```

        PWMH = NILAIHP + NILAIHD + NILAIHI + 1.2
'Control PID Horizontal
    If PWMH > 3 Then
        PWMH = 3
    End If

    TextBox12.Text = PWMH
    AO1.Write(1, PWMH)

    If DataIn1 < 0.5 Then                                'proximity
Vertikal
        a = a
        b = b
        c = 0
        d = 0
    End If

    TextBox15.Text = NILAIVP
    TextBox16.Text = NILAIVD
    TextBox17.Text = NILAIVI
    TextBox18.Text = NILAIHP
    TextBox19.Text = NILAIHD
    TextBox20.Text = NILAIHI

End Sub
Private Sub tambah_titik(ByVal input_x, ByVal
input_y)
    Dim removedpoint As Boolean = False
    Dim radiussquare As Double = radius ^ 2
    removedpoint = False

    'menambah titik baru...
    If Not removedpoint Then
        puntiBtemp(0) = New PointF() {New
PointF(input_x, input_y)}
        add_remove_pt = True
    End If

```

```

End Sub
Private Sub lukas()

    frame = citraasli
    grayB = frame.Convert(Of Emgu.CV.Structure.Gray,
Byte)()

    If defineCorners Then

        count = 500
        puntiB = grayB.GoodFeaturesToTrack(count,
0.01, 10, 3)
        grayB.FindCornerSubPix(puntiB, New
System.Drawing.Size(10, 10), New System.Drawing.Size(-1,
-1), New Emgu.CV.Structure.MCvTermCriteria(20, 0.03))
        count = puntiB(0).Length
        defineCorners = False
    End If

    If count > 0 Then

        Emgu.CV.OpticalFlow.PyrLK(grayA, grayB,
pyrBufferA, pyrBufferB, puntiA(0), New Size(10, 10), 5,
New Emgu.CV.Structure.MCvTermCriteria(20, 0.03D), flags,
puntiB(0), status, errors)
        flags =
Emgu.CV.CvEnum.LKFLOW_TYPE.CV_LKFLOW_PYR_A_READY

        Dim k As Integer = 0
        Dim col2 As New Emgu.CV.Structure.Bgr(255,
128, 0)

        For i As Integer = 0 To count - 1

            If status(i) > 0 Then

                Dim cr As New
Emgu.CV.Structure.CircleF(puntiB(0)(i), 4)
                puntiB(0)(k) = puntiB(0)(i)
                frame.Draw(cr, col2, 2)
            End If
        Next i
    End If
End Sub

```

```

        k += 1

    End If
Next

    count = k
    posisiX = Int(puntiB(0)(0).X)
    posisiY = Int(puntiB(0)(0).Y)
    errorx1 = posisiX - 320
    errory1 = posisiY - 240
End If

If add_remove_pt Then

    ' grayB.FindCornerSubPix(puntiBtemp, New
System.Drawing.Size(10, 10), New System.Drawing.Size(-1,
-1), New Emgu.CV.Structure.MCvTermCriteria(20, 0.03))

    If count > 0 Then
        ReDim Preserve
puntiB(0)(puntiB(0).Length)
        puntiB(0)(puntiB(0).Length - 1) =
puntiBtemp(0)(0)
    Else
        puntiB(0) = puntiBtemp(0)
    End If

    count = puntiB(0).Length
    add_remove_pt = False

End If
' swap images:
imgTemp = grayA
grayA = grayB
grayB = imgTemp
'swap pyramid
imgTemp = pyrBufferA
pyrBufferA = pyrBufferB
pyrBufferB = imgTemp
' swap points:

```

```

        puntiTemp = puntiA
        puntiA = puntiB
        puntiB = puntiTemp
    End Sub
    Private Sub FaceDetection()

        imGray = citraasli.Convert(Of Gray, Byte)()

        faceDetected = imGray.DetectHaarCascade(face,
1.1, 3, CvEnum.HAAR_DETECTION_TYPE.DO_ROUGH_SEARCH, New
Size(30, 30))(0)

        xx = 0
        yy = 0
        For Each fo In faceDetected
            imGray.ROI = fo.rect
            xx = (fo.rect.X + (fo.rect.Width / 2))
            yy = (fo.rect.Y + (fo.rect.Height / 2))
            roi_wajah = New Rectangle(fo.rect.X,
fo.rect.Y, ((fo.rect.Width)), fo.rect.Height)
            bunder = New Rectangle((fo.rect.X +
(fo.rect.Width / 2)), (fo.rect.Y + (fo.rect.Height /
2))), 5, 5)
            imGray2.ROI = bunder
            imGray.ROI = roi_wajah
            citraasli.Draw(fo.rect, New Bgr(Color.Red),
1)
            citraasli.Draw(bunder, New Bgr(Color.Aqua),
2)
            imGray.ROI = Rectangle.Empty
            imGray2.ROI = Rectangle.Empty
        Next
    End Sub
    Private Sub PictureBox1_MouseDown(ByVal sender As
Object, ByVal e As System.Windows.Forms.MouseEventArgs)
Handles PictureBox1.MouseDown

        If e.Button = MouseButton.Left Then
            tambah_titik(e.X, e.Y)
        End If
    End Sub

```

```

End Sub

Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
    Application.Exit()
End Sub

Private Sub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BUTTON2.Click
    If BUTTON2.Text = "START" Then

capdev.SetCaptureProperty(CvEnum.CAP_PROP.CV_CAP_PROP_FR
AME_WIDTH, 640)

capdev.SetCaptureProperty(CvEnum.CAP_PROP.CV_CAP_PROP_FR
AME_WIDTH, 480)

        Timer1.Enabled = True
        BUTTON2.Text = "STOP"
    Else
        Timer1.Enabled = False
        BUTTON2.Text = "START"
    End If
End Sub

Private Sub Button3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button3.Click

    a = 0
    b = 0
    c = 0
    d = 0
    NILAIVP = 0
    NILAIVD = 0

```

```

NILAIVI = 0
NILAIHP = 0
NILAIHD = 0
NILAIHI = 0

PWMH = 0
PWMV = 0
XP = 0
XI = 0
YP = 0
YI = 0

errorry1 = 0
errorx1 = 0
errorx = 0
errorry = 0
posisiX = 0
posisiY = 0

For i As Integer = 0 To count - 1

    If status(i) > 0 Then

        Dim cr As New
Emgu.CV.Structure.CircleF(puntiB(0)(i), 4)

        puntiB(0)(i).X = 0
        puntiB(0)(i).Y = 0

    End If
Next
count = 0
End Sub

Private Sub Button4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
    If tembak = 0 Then
        Button4.Text = "BERHENTI"
        tembak = 1
    End If
End Sub

```

```

Else
    tembak = 0
    Button4.Text = "TEMLAK"
End If

End Sub

Private Sub Button5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
    AO0.Write(0, manualH)
    DigitalOut.Write(0, 2)
End Sub

Private Sub Button8_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
    AO0.Write(0, 0)
    AO1.Write(1, 0)
    DigitalOut.Write(0, 0)

End Sub

Private Sub Button6_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button6.Click
    AO1.Write(1, manualL)
    DigitalOut.Write(0, 4)
End Sub

Private Sub Button9_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button9.Click
    AO0.Write(0, manualH)
    DigitalOut.Write(0, 1)
End Sub

Private Sub Button7_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button7.Click

```

```
        AO1.Write(1, manualL)  
        DigitalOut.Write(0, 8)  
    End Sub  
End Class
```




FINAL PROJECT - TE 141599

**DESIGN AND REALIZATION OF AUTO TRACKING AND
TARGET LOCK ON STATIC DEFENSE BASED ON IMAGE
PROCESSING**

Dani Prasetyawan
NRP 2211100011

Supervisor
Rudy Dikairono, ST., MT.
Ronny Mardiyanto, ST., MT., Ph.D.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2015

RANCANG BANGUN SISTEM PELACAKAN OTOMATIS DAN PENGUNCIAN SASARAN PADA PERTAHANAN STATIS BERBASIS PENGOLAH CITRA

Nama : Dani Prasetyawan
Pembimbing I : Rudy Dikairono, ST., MT.
Pembimbing II : Ronny Mardiyanto, ST., MT, Ph.D.

ABSTRAK

Sistem pelacakan otomatis dan penguncian sasaran merupakan sebuah sistem senjata yang diotomasi. Sistem tersebut bekerja dengan melacak dan mengunci sasaran secara otomatis terhadap target yang telah dipilih terlebih dahulu. Sistem ini dapat menggantikan peran manusia pada suatu pertahanan titik. Sistem seperti ini yang ada sebelumnya menggunakan teknologi radar dan *opto-electrical*. Teknologi tersebut rentan *jamming* dan memiliki tingkat kesusahan yang tinggi serta biaya pengadaan yang mahal sehingga digunakanlah sistem serupa dengan basis penginderaan visual menggunakan kamera.

Kamera diletakkan pada sistem ini bertujuan untuk proses pengambilan data *visual*. Dari data *visual* tersebut dipilih target untuk diproses. Kemudian data yang didapat diproses oleh *processing unit* dengan tujuan untuk dapat melacak sasaran dan mengunci sasaran yang telah dipilih. Berikutnya *operator* dapat mengambil keputusan apakah sasaran perlu ditembak atau dibiarkan.

Hasil dari pengujian yang dilakukan pada tugas akhir ini adalah sistem dapat melacak sasaran berupa manusia dengan sasaran dapat berupa sasaran bergerak maupun sasaran diam yang dapat ditentukan operator. Lama waktu pelacakan dapat melebihi 40detik untuk sasaran bergerak. Untuk *error* penguncian sasaran adalah 10% dari lebar total *frame image* baik untuk sumbu tinggi maupun lebar. Selain itu rata-rata akurasi penembakan yang didapat dari pengujian penembakan dengan jarak dan ketinggian sasaran yang bervariasi adalah 50%.

Kata kunci : Kamera, penginderaan visual, *processing unit*



Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas kasih dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir ini dengan judul :

Rancang Bangun Sistem Pelacakan Otomatis dan Penguncian Sasaran Pada Pertahanan Statis Berbasis Pengolah Citra

Tugas Akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dibuat berdasarkan teori-teori yang didapat selama mengikuti perkuliahan, berbagai literatur penunjang dan pengarahan dosen pembimbing dari awal hingga akhir pengerjaan Tugas Akhir ini.

Pada kesempatan ini, penulis ingin berterima kasih kepada pihak-pihak yang membantu pembuatan tugas akhir ini, khususnya kepada:

1. Bapak, Ibu, adik serta seluruh keluarga yang memberikan dukungan baik moril maupun materiil.
2. Rudy Dikairono S.T., M.T. selaku dosen pembimbing 1 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
3. Ronny Mardiyanto, S.T., M.T.,P.hd. selaku dosen pembimbing 2 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
4. Tasripan, IR. MT selaku Koordinator Bidang Studi Elektronika.
5. Dr. Tri Arief Sardjono, ST, MT selaku Ketua Jurusan Teknik Elektro ITS Surabaya.
6. Seluruh dosen bidang studi elektronika.
7. Teman-teman laboratorium Elektronika yang tidak dapat disebutkan satu-persatu, telah membantu proses pengerjaan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini belum sempurna dan masih banyak hal yang dapat diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis untuk pengembangan lebih lanjut.

Akhirnya penulis berharap agar Tugas Akhir ini dapat memberikan manfaat yang sebesar-besarnya bagi semua pihak serta pengembangan aplikasi *image processing*. Penulis juga berharap supaya Tugas Akhir ini dapat menjadi aplikasi yang lebih bermanfaat.

Surabaya, 11 Mei
2015

Penulis

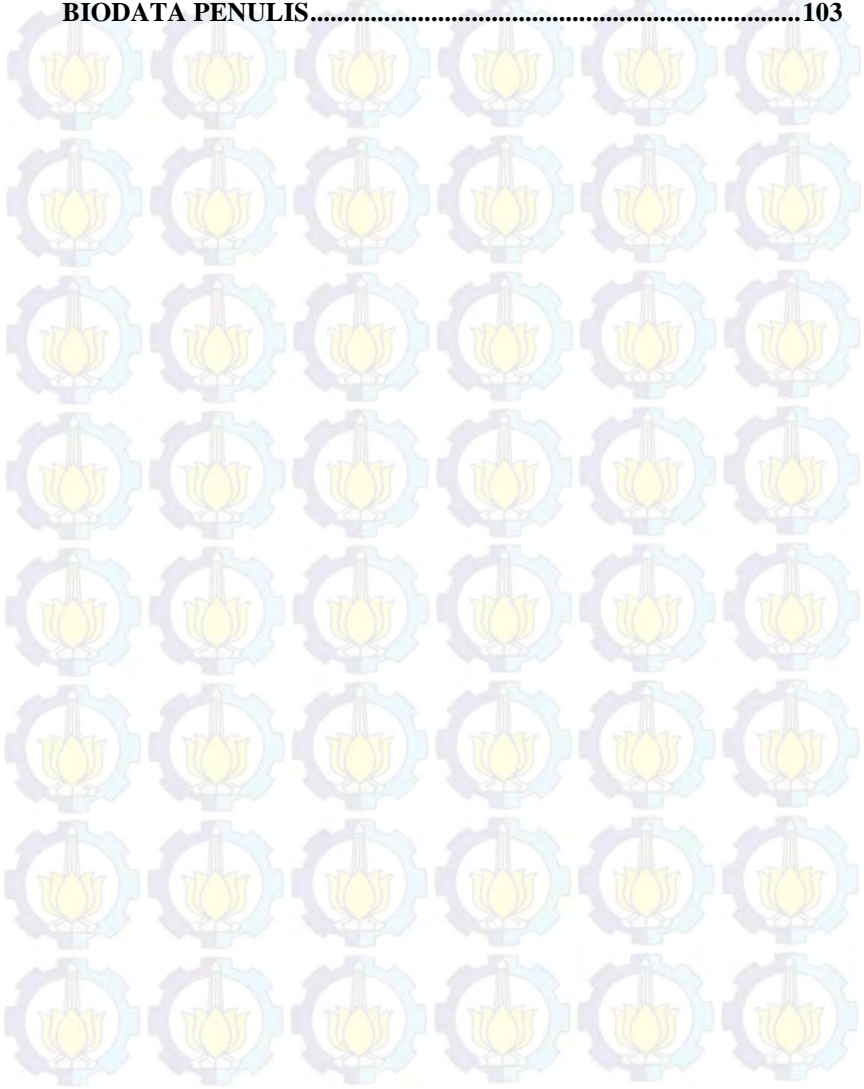


DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi	3
1.6 Sistematika Penulisan.....	5
1.7 Relevansi	5
BAB II TINJAUAN PUSTAKA DAN TEORI PENUNJANG.....	7
2.1 Citra Digital.....	7
2.2.1 Citra Biner.....	8
2.2.2 Citra Warna (RGB)	9
2.2.3 <i>Grayscale</i>	10
2.2 <i>Image Processing</i>	12
2.2.1 <i>Optical Flow</i>	12
2.2.1.1 Gambaran <i>Optical Flow</i>	12
2.2.1.2 Algoritma <i>Lucas-Kanade</i>	12
2.2.1.3 <i>Pyramidal Lukas-Kanade</i>	12
2.2.2 <i>Template Matching</i>	13
2.2.3 <i>Human Detection</i>	14
2.2.4 Emgu CV	14
2.2.5 Open CV	15
2.3 Jenis Senjata	15
2.3.1 <i>Machine Gun</i>	15
2.3.2 <i>Submachine Gun</i>	16
2.3.3 <i>Assault Rifle</i>	16
2.4 Data Acquisition Module.....	17

2.5	Pulse Width Modulation	18
2.6	Driver Motor	18
2.7	Gerak Peluru	20
2.8	Penelitian Terkait	21
2.8.1	CIWS (Close In Weapon System)	21
BAB III PERANCANGAN SISTEM.....		23
3.1	Perancangan <i>Software</i>	27
3.1.1	Pengambilan Citra	27
3.1.2	<i>Grayscale</i>	28
3.1.3	Optical Flow	30
3.1.4	<i>Haar cascade</i>	32
3.1.5	Kontrol PID	33
3.1.6	Koneksi <i>Advantech DAQ Module</i>	34
3.2	Hardware	34
3.2.1	<i>Advantech USB-4711A</i>	34
3.2.2	Sensor <i>Proximity</i>	36
3.2.3	Rangkaian Pembangkit PWM (<i>Pulse Width Modulation</i>)	37
3.2.4	Aktuator Tembak	40
3.2.5	<i>Driver Motor</i>	41
3.2.6	Motor DC	43
3.2.7	Senjata	44
3.2.8	Kamera	45
3.2.9	Komputer	45
3.2.10	Sistem Penembak	46
BAB IV PENGUJIAN.....		49
4.1	Pengujian Rangkaian Pembangkit PWM	49
4.2	Pengujian Motor	53
4.3	Pengujian Pendeteksi Manusia	54
4.4	Pengujian Pelacakan dan Penguncian Sasaran pada Siang Hari	57
4.5	.Pengujian Pelacakan dan Penguncian Sasaran di Malam Hari	67
4.6	Pengujian Penembakan	74
BAB V KESIMPULAN DAN SARAN.....		81
	Kesimpulan	81
	Saran	81

DAFTAR PUSTAKA83
LAMPIRAN.....85
BIODATA PENULIS.....103





Halaman ini sengaja dikosongkan

DAFTAR TABEL

	Halaman
Tabel 3.1 Fungsi Penggunaan Port Advantech USB-4711	36
Tabel 3.2 Tabel kebenaran driver motor Vertikal.....	42
Tabel 3.3 Tabel kebenaran driver motor Horizontal.....	42
Tabel 4.1 Pengukuran keluaran <i>Duty Cycle</i> terhadap tegangan referensi	50
Tabel 4.2 Pengujian motor DC vertikal	53
Tabel 4.3 Pengujian motor DC horizontal	54
Tabel 4.4 Tabel pengujian deteksi manusia.....	55
Tabel 4.5 Tabel pengujian pelacakan pada siang hari	58
Tabel 4.6 Tabel alur gerak sasaran dan keterangan pengujian pelacakan pada siang hari.....	58
Tabel 4.7 Tabel pengujian pelacakan kedua pada siang hari purwarupa SIG552	61
Tabel 4.8 Tabel pengujian pelacakan kedua pada siang hari purwarupa M4	61
Tabel 4.9 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada siang hari purwarupa SIG552	62
Tabel 4.10 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada siang hari purwarupa M4.....	63
Tabel 4.11 Skenario pengujian gangguan sasaran	65
Tabel 4.12 Tabel pengujian pelacakan pada malam hari 1	67
Tabel 4.13 Tabel alur gerak sasaran dan keterangan pengujian pelacakan pada siang hari 1.....	68
Tabel 4.14 Tabel pengujian pelacakan kedua pada malam hari purwarupa SIG552	70
Tabel 4.15 T abel pengujian pelacakan kedua pada malam hari purwarupa M4	70
Tabel 4.16 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada malam hari purwarupa SIG552	71
Tabel 4.17 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada malam hari purwarupa M4	72
Tabel 4.18 Tabel pengujian tembak.....	75
Tabel 4.19 Tabel hasil pengujian tembak	75



Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

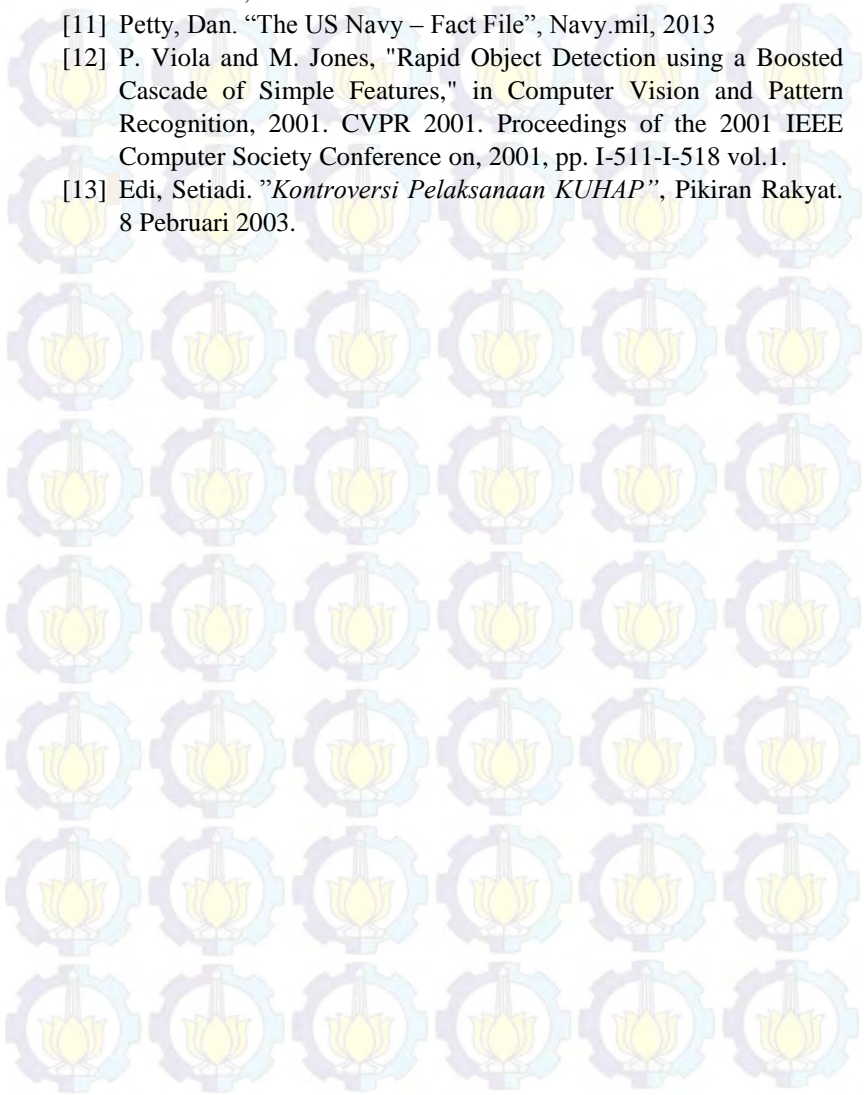
	Halaman
Gambar 2.1 Posisi koordinat digital	8
Gambar 2.2 Citra Biner.....	9
Gambar 2.3 Representasi warna RGB pada citra digital	10
Gambar 2.4 Citra hasil operasi <i>grayscale</i>	11
Gambar 2.5 <i>Pyramidal Optical Flow</i>	13
Gambar 2.6 Logo EmguCV	14
Gambar 2.7 Logo OpenCV[8]	15
Gambar 2.8 M249 <i>Light Machine Gun</i>	16
Gambar 2.9 MP5 <i>Submachine Gun</i>	16
Gambar 2.10 M4 <i>Assault Rifle</i>	17
Gambar 2.11 Advantech Portable Data Acquisition Module [9].....	17
Gambar 2.12 Presentase <i>Duty Cycle</i> pada PWM.....	18
Gambar 2.13 Konfigurasi rangkaian <i>H-Bridge</i>	19
Gambar 2.14 Lintasan gerak peluru.....	20
Gambar 2.15 Phalanx CIWS[11].....	21
Gambar 3.1 Skenario pemilihan target	23
Gambar 3.2 Skenario sistem penembak mengikuti pergerakan target	24
Gambar 3.3 Diagram Blok Sistem	25
Gambar 3.4 <i>Flowchart</i> Sistem.....	26
Gambar 3.5 Tampilan <i>software</i>	27
Gambar 3.6 Citra frame	28
Gambar 3.7 Video hasil <i>capture</i> dalam RGB diubah ke <i>Grayscale</i>	28
Gambar 3.8 <i>Flowchart grayscale</i>	29
Gambar 3.9 Penandaan sasaran oleh program	31
Gambar 3.10 Diagram blok sistem kontrol PID	34
Gambar 3.11 Konfigurasi port Advantech USB-4711A	35
Gambar 3.12 Advantech USB-4711A	36
Gambar 3.13 Sensor <i>Proximity</i>	37
Gambar 3.14 Skematik rangkaian pembangkit PWM	38
Gambar 3.15 Rangkaian pembangkit PWM (dalam kotak)	39
Gambar 3.16 Skematik rangkaian aktuator tembak	40
Gambar 3.17 Rangkaian aktuator menembak	41

Gambar 3.18 Driver motor BTN7870B	42
Gambar 3.19 Motor DC penggerak vertikal	43
Gambar 3.20 Motor DC penggerak horizontal	43
Gambar 3.21 Purwarupa M4 1:1	44
Gambar 3.22 Purwarupa SIG552 1:2	44
Gambar 3.23 Logitech C920	45
Gambar 3.24 Rancangan sistem penembak dengan purwarupa SIG552	46
Gambar 3.25 Rancangan sistem penembak dengan purwarupa M4	47
Gambar 3.26 Posisi motor DC penggerak horizontal	48
Gambar 4.1 Pengujian rangkaian pembangkit PWM	49
Gambar 4.2 Grafik hasil pengukuran keluaran <i>duty cycle</i> terhadap tegangan referensi	51
Gambar 4.3 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 0,8Volt	52
Gambar 4.4 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 2,2Volt	52
Gambar 4.5 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 3,6Volt	52
Gambar 4.6 Pengujian pendeteksi manusia	55
Gambar 4.7 Pengujian deteksi manusia	56
Gambar 4.8 Pengujian deteksi manusia	56
Gambar 4.9 Pengujian deteksi manusia	57
Gambar 4.10 Proses pengujian pelacakan pada siang hari	60
Gambar 4.11 Pengujian gangguan di depan sasaran sebelum gangguan	66
Gambar 4.12 Pengujian gangguan di depan sasaran sesudah gangguan menutupi sasaran	66
Gambar 4.13 Proses pengujian pelacakan pada malam hari	73
Gambar 4.14 Grafik Akurasi Penembakan Sasaran	78
Gambar 4.15 Proses pengujian penembakan	79

DAFTAR PUSTAKA

- [1] Johnny Andrean Susanto, Kelvin Giovanni Lukman, "Digital Watermarking Untuk Melindungi Informasi Multimedia Dengan Metode Fast Fourier Transform (FFT)", <URL:http://eprints.mdp.ac.id/891/1/JURNAL%202009250032%20JOHNNY_ANDREAN_SUSANTO%20DAN%202009250027%20KELVIN_GIOVANNI_LUKMAN.pdf>,2003.
- [2] _____, "Teori dasar citra digital",<URL:<http://repository.usu.ac.id/bitstream/123456789/16877/4/Chapter%20II.pdf>>
- [3] Putra, Darma, "Pengolahan Citra Digital", Yogyakarta, Penerbit ANDI, 2010.
- [4] Rachmawati, "Estimasi Parameter Geometris Benda Berbasis Pengolahan Citra Digital", <URL: http://digilib.telkomuniversity.ac.id/index.php?option=com_content&view=article&id=344:citra-digital&catid=15:pemrosesan-sinyal&Itemid=14>, Desember, 2008.
- [5] Achmad Rizal, "Pengolahan Citra", <URL: <http://achmadrizal.staff.telkomuniversity.ac.id/2014/06/19/pengolahan-citra/>>, Juni, 2014.
- [6] Ubaidillah Umar, Reni Soelistijorini, Haryadi Amran Darwito, "Tracking Arah Gerakan Telunjuk Jari Berbasis Webcam Menggunakan Metode Optical Flow", Electronic Engineering Polytechnic Institute of Surabaya (EEPIS), Indonesia, October 26, 2011.
- [7] Indra Pramana, M Zen Hadi Samson, Setiawardhana, "TRACKING OBJECT MENGGUNAKAN METODE TEMPLATE MATCHING BERBASIS STEREO VISION", Surabaya.
- [8] Bradski, Gary dan Kaebler, Adrian. "*Learning OpenCV Computer Vision with the OpenCV Library*", O' REILLY. 2008.
- [9] _____, "Advantech USB-4711A Module", <URL: http://www2.advantech.eu/products/1-2MLKNO/USB4711A/mod_16CB666D-62EE-4C15-BF5F-EA49F61093CF.aspx>

- [10] Williams, Al “*Microcontroller projects using the Basic Stamp*” Edisi kedua, *Focal Press*.
- [11] Petty, Dan. “The US Navy – Fact File”, Navy.mil, 2013
- [12] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001, pp. I-511-I-518 vol.1.
- [13] Edi, Setiadi. “*Kontroversi Pelaksanaan KUHAP*”, Pikiran Rakyat. 8 Pebruari 2003.



BIODATA PENULIS



Dani Prasetyawan lahir di Jombang pada 10 Agustus 1992. Anak pertama dari 2 bersaudara dari pasangan Djoko Purwanto dan Rivana Harijani. Penulis menyelesaikan pendidikan dasar di SDN Penjaringan Sari II Surabaya, Surabaya dilanjutkan dengan pendidikan menengah di SMPN 1 Surabaya dan SMAN 5 Surabaya. Pada tahun 2011, penulis memulai pendidikan di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi

Sepuluh Nopember (ITS) Surabaya. Selama kuliah penulis aktif membantu penyelenggaraan kegiatan dan aktif sebagai asisten laboratorium Elektronika Dasar dan praktikum Elektronika maupun koordinator asisten pada semester ganjildan genap 2014-2015.

Email :

daniprasetyawan1992@gmail.com



Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam bidang militer, penggunaan tenaga manusia untuk melakukan tugas yang membutuhkan tingkat akurasi dan presisi yang tinggi perlahan dikurangi dan digantikan dengan sistem yang dapat berjalan secara mandiri. Contoh dari hal yang membutuhkan akurasi dan presisi yang tinggi adalah penajakan terhadap target dan penguncian terhadap target.

Seiring berkembangnya teknologi senjata, senjata yang digunakan semakin canggih, semakin cepat, dan sulit untuk dideteksi dan dikunci secara cepat dan akurat dengan kemampuan manusia. Manusia memiliki batasan untuk melihat benda dengan kecepatan tertentu dan pada jarak tertentu, sehingga membutuhkan bantuan alat untuk dapat menutupi kekurangan itu. Disamping itu, kemampuan manusia juga memiliki batasan berupa tidak dapat fokus pada suatu hal secara terus menerus diakibatkan memiliki batasan tenaga dan kelelahan akibat dari konsentrasi secara berkala. Hal tersebut dapat mengakibatkan kesalahan dalam pengambilan keputusan berupa keputusan untuk melakukan penembakan yang tidak tepat sasaran terhadap target yang telah dipilih. Hal ini sangat merugikan terutama untuk bidang militer.

Baru-baru ini senjata yang telah diotomasi pada beberapa aspek sedang sangat diminati karena dapat mencegahnya terjadinya *human error* maupun korban manusia dan dapat diandalkan. Senjata model seperti ini juga dapat mencegah atau mengurangi terjadinya korban yang tidak diinginkan akibat terjadinya *human error* karena tidak membutuhkan *operator* yang bekerja di tempat dan mengurangi tingkat akumulasi kelelahan pada *operator*.

Selain itu peperangan asimetris akhir-akhir ini menyebabkan banyak korban. Contohnya adalah bom bunuh diri yang ditempel pada tubuh pegebom bunuh diri menjadi salah satu penyumbang korban perang yang besar. Berdasar data oleh Dr Madelyn Hsiao-Rei Hicks, Institute of Psychiatry, King's College London, UK, korban bom bunuh diri di Iraq antara tahun 2003 hingga 2010 mencapai lebih dari 12.000 orang sipil, dan 300 personel militer.

Oleh karena itu dalam tugas akhir ini diimplementasikan rancang bangun sistem pelacakan otomatis dan penguncian sasaran untuk pertahanan statis berbasis pengolah citra. Dengan menggunakan teknik pengolah citra (*Image Processing*) yang dilakukan oleh *Processing Unit* dapat dilakukan penjajakan dan penguncian sasaran secara otomatis. Diharapkan agar sistem ini dapat menutupi kekurangan yang ada pada manusia untuk melakukan tugas serupa dan dapat mencegah timbulnya korban yang tidak diinginkan maupun terjadinya kesalahan yang terjadi akibat keterbatasan yang dimiliki oleh manusia.

1.2 Perumusan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini adalah:

1. Bagaimana mendeteksi target yang ada?
2. Bagaimana pelacakan target dapat dilakukan?
3. Bagaimana penguncian target dapat dilakukan?
4. Bagaimana merencanakan pergerakan sistem untuk mengikuti pergerakan target?

1.3 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah:

1. Pergerakan motor horizontal terdapat delay akibat sistem mekanik.
2. Motor vertikal terdapat tambahan beban akibat gaya gravitasi.
3. Lokasi pelaksanaan memiliki pencahayaan yang cukup.
4. Akurasi dan jarak penembakan bergantung jenis senjata.
5. Jarak deteksi minimal 3 Meter.
6. Target dipilih secara manual.

1.4 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Sistem mampu melacak target secara tepat.
2. Sistem mampu mengikuti pergerakan target secara tepat.
3. Sistem mampu bekerja dengan kecepatan yang bervariasi.
4. Dapat mengkomunikasikan hasil pemrosesan data citra hasil pelacakan dan penguncian kepada *operator* secara baik.

5. Dapat menempatkan kamera, *processing unit*, dan sistem senjata secara baik agar tidak membebani *operator*.

1.5 Metodologi

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut :

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Tugas Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, dan artikel-artikel di internet dan forum-forum diskusi internet.

2. Perancangan Software

Perangkat lunak dirancang dengan pembuatan *source code* melakukan pelacakan target, penguncian target dan penembakan target. Untuk pemilihan sasaran, dilakukan oleh *operator* secara *manual*. Untuk pelacakan dan penguncian sasaran, dapat dilakukan dengan membandingkan *error* dari jarak target sekarang menuju titik tengah (titik tembak) dan *error* pada *frame* dari *image* yang diterima saat itu dengan *error* pada *frame* sebelumnya yang ditangkap oleh kamera yang digunakan sebagai referensi sasaran. Lalu dilakukan perhitungan agar sasaran selalu berada di tengah dari hasil *image* yang diterima oleh kamera. Untuk melakukannya, diperlukan *source code* yang akan digunakan untuk mengirim perintah pada *hardware* agar bergerak mengikuti sesuai arahan dari *software*. Terakhir, dibuat *code* agar jika sasaran melebihi batasan dari sistem yang dibuat, maka sistem akan otomatis berhenti mengejar sasaran.

3. Perancangan Hardware

Perancangan *hardware* pada tahap ini meliputi kamera, *data acquisition module* untuk menggerakkan *hardware* berupa motor untuk pergerakan *pan-tilt*, danudukan

senjata serta kamera. Motor, kamera, dan senjata akan dihubungkan dengan *processing unit* agar dapat dikendalikan pergerakannya. Untuk kedudukan sistem dibuat tetap, tidak bergerak.

4. Pengujian Sistem

Pengujian alat dilakukan untuk menentukan keandalan dari sistem yang telah dirancang. Pengujian dilakukan untuk melihat apakah *software* dan *hardware* dapat bekerja secara baik.

Pengujian sistem dilakukan dengan pelacakan dan penguncian target atau sasaran yang bergerak yang telah dipilih. Apakah sistem dapat mengikuti pergerakan target tersebut dengan cara selalu mengarahkan senjata menuju target atau tidak. Lalu dilakukan pengujian tembak apakah tembakan dari senjata yang diarahkan mengenai sasaran yang telah dikunci tersebut

5. Analisa

Analisa dilakukan terhadap hasil dari pengujian sehingga dapat ditentukan karakteristik pelacakan dan penguncian dari *software* dan *hardware* yang telah dibuat. Apabila karakteristik pelacakan dan penguncian dari *software* dan *hardware* yang telah dibuat masih belum sesuai, maka perlu dilakukan perancangan ulang pada sistem dan diuji kembali.

6. Penulisan Laporan Tugas Akhir

Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, tinjauan pustaka dan teori penunjang, perancangan sistem, pengujian, dan penutup.

1.6 Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

➤ BAB I : PENDAHULUAN

Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

➤ BAB II : TINJAUAN PUSTAKA DAN TEORI PENUNJANG

Bab ini menjelaskan tentang teori penunjang dan *literature* yang dibutuhkan dalam pengerjaan tugas akhir ini. Dasar teori yang menunjang meliputi citra digital, *haarcascade upper body detection*, *optical flow*, *Pulse Width Modulation*, *Data Acquisition Module*, *driver motor*, dan Motor DC. Bagian ini memaparkan mengenai beberapa teori penunjang dan beberapa literatur yang berguna bagi pembuatan Tugas Akhir ini.

➤ BAB III : PERANCANGAN SISTEM

Bab ini menjelaskan tentang perencanaan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk sistem pelacakan dan penguncian target secara otomatis.

➤ BAB IV : PENGUJIAN

Pada bab ini akan menjelaskan hasil uji coba sistem beserta analisisnya.

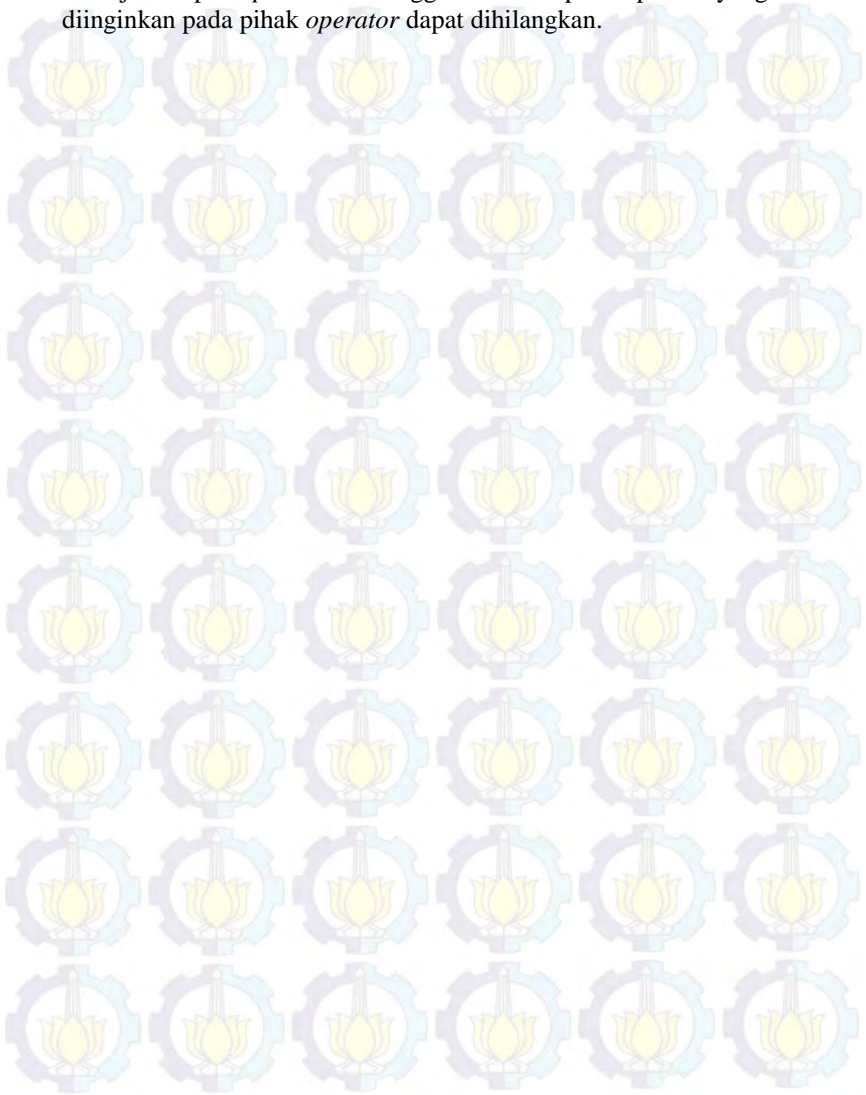
➤ BAB V : PENUTUP

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran-saran untuk pengembangan lebih lanjut.

1.7 Relevansi

Hasil yang diharapkan dari tugas akhir ini diharapkan mampu meringankan pekerjaan manusia khususnya pada bidang militer khususnya untuk pertahanan statis. Pengembangan lebih lanjut dari

sistem ini adalah penambahan penembakan otomatis agar sistem dapat bekerja tanpa *operator* sehingga korban pada pihak yang tidak diinginkan pada pihak *operator* dapat dihilangkan.



BAB II

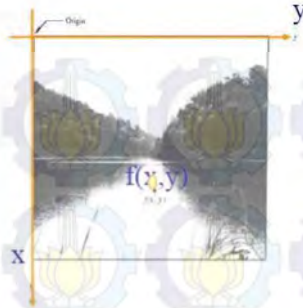
TINJAUAN PUSTAKA DAN TEORI PENUNJANG

Teori penunjang dalam bab ini menjelaskan tentang teori penunjang yang berhubungan dengan keseluruhan sistem yang akan dibuat pada tugas akhir ini. Sedangkan tinjauan pustaka dalam bab ini menjelaskan tentang sistem-sistem yang berhubungan dengan tugas akhir ini dan pernah diimplementasikan oleh penulis-penulis sebelumnya.

2.1 Citra Digital

Citra digital adalah gambar dua dimensi yang dapat ditampilkan pada layar monitor komputer sebagai himpunan berhingga (diskrit) dengan nilai digital yang disebut dengan *pixel* (*picture elements*). *Pixel* adalah elemen citra yang memiliki nilai yang menunjukkan intensitas warna. Citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar/ pixel/ piksel/ pels/ picture element) menyatakan tingkat keabuan pada titik tersebut. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi.

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial pada citra tersebut dan nilai $f(x,y)$ yang merupakan intensitas citra pada posisi koordinat tersebut. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*) dengan perubahan intensitas dari ketiga warna dasar tersebut menghasilkan intensitas pewarnaan yang berbeda. Sistem koordinat yang ada pada sebuah citra digital dapat dilihat pada contoh Gambar 2.1[1] berikut.



Gambar 2.1 Posisi koordinat digital

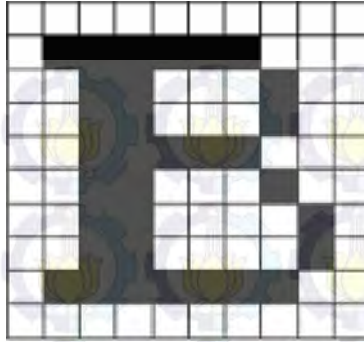
Berdasarkan cara penyimpanan atau pembentukannya, citra digital dapat dibagi menjadi dua jenis. Jenis pertama adalah citra digital yang dibentuk oleh kumpulan pixel dalam *array* dua dimensi. Citra jenis ini disebut citra bitmap (*bitmap image*) atau citra raster (*raster image*). Jenis citra yang kedua adalah citra yang dibentuk oleh fungsi-fungsi geometri dan matematika. Jenis citra ini disebut grafik vektor (*vector graphics*).

Citra digital (diskrit) dihasilkan dari citra analog (kontinu) melalui digitalisasi. Digitalisasi citra analog terdiri atas *sampling* dan kuantisasi (*quantization*). *Sampling* adalah pembagian citra ke dalam elemen-elemen diskrit (pixel), sedangkan kuantisasi adalah pemberian nilai intensitas warna pada setiap pixel dengan nilai yang berupa bilangan bulat (G.W. Awcock, 1996).

Banyaknya nilai yang dapat digunakan dalam kuantisasi citra bergantung kepada kedalaman pixel, yaitu banyaknya bit yang digunakan untuk merepresentasikan intensitas warna pixel. Kedalaman pixel sering disebut juga kedalaman warna. Citra digital yang memiliki kedalaman pixel n bit disebut juga citra n -bit.[2]

2.2.1 Citra Biner

Citra biner adalah citra dimana piksel-pikselnya hanya memiliki dua buah nilai intensitas yaitu bernilai 0 dan 1 dimana 0 menyatakan warna latar belakang (*background*) dan 1 menyatakan warna tinta/objek atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih. Citra biner diperoleh dari nilai citra *threshold* sebelumnya. Gradasi citra biner dapat dilihat pada Gambar 2.2. [3]



Gambar 2.2 Citra Biner

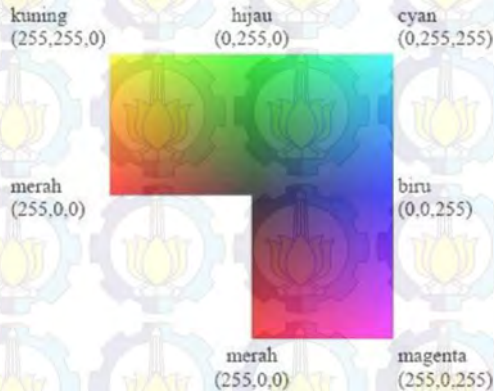
Meskipun saat ini citra berwarna lebih disukai karena memberi kesan warna yang lebih kaya dari pada citra biner, namun tidak membuat eksistensi citra biner mati. Pada beberapa aplikasi citra biner masih tetap di butuhkan, misalkan citra logo instansi (yang hanya terdiri dari warna hitam dan putih), citra kode barang (*bar code*) yang tertera pada label barang, citra hasil pemindaian dokumen teks, dan sebagainya. Seperti yang sudah disebutkan diatas, citra biner hanya mempunyai dua nilai derajat keabuan, hitam dan putih. Pixel – pixel objek bernilai 1 dan pixel – pixel latar belakang bernilai 0. Pada waktu menampilkan gambar, adalah putih dan 1 adalah hitam. Jadi pada citra biner, latar belakang berwarna putih sedangkan objek berwarna hitam seperti tampak pada gambar 2.2 diatas. Meskipun komputer saat ini dapat memproses citra hitam-putih (*grayscale*) maupun citra berwarna, namun citra biner masih tetap di pertahankan keberadaannya.

Alasan penggunaan citra biner adalah karena citra biner memiliki sejumlah keuntungan seperti kebutuhan memori kecil karena nilai derajat keabuan hanya membutuhkan representasi 1 bit dan Waktu pemrosesan lebih cepat di bandingkan dengan citra hitam-putih ataupun warna.

2.2.2 Citra Warna (RGB)

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang nilai. Untuk

monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak $256 \times 256 \times 256 = 16777216$ jenis warna.



Gambar 2.3 Representasi warna RGB pada citra digital

Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang dimensi 3 yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalkan sebuah vektor dituliskan sebagai $r = (x,y,z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen *Red*, *Green*, *Blue*. Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = $RGB(30, 75, 255)$. Putih = $RGB(255,255,255)$, sedangkan untuk hitam = $RGB(0,0,0)$. [4]

2.2.3 Grayscale

Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih" saja. Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra *grayscale* seringkali merupakan

perhiungan dari intensitas cahaya pada setiap pixel pada spektrum elektromagnetik single band.

Citra *grayscale* disimpan dalam format 8 bit untuk setiap sample piksel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi citra *grayscale* dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B. [5]

Citra *grayscale* merupakan citra satu kanal, dimana citra $f(x,y)$ merupakan fungsi tingkat keabuan dari hitam keputih, x menyatakan variable kolom atau posisi pixel di garis jelajah dan y menyatakan variable kolom atau posisi pixel di garis jelajah. Intensitas f dari gambar hitam putih pada titik (x,y) disebut derajat keabuan (*grey level*), yang dalam hal ini derajat keabuannya bergerak dari hitam keputih. Derajat keabuan memiliki rentang nilai dari I_{min} sampai I_{max} , atau $I_{min} < f < I_{max}$, selang (I_{min} , I_{max}) disebut skala keabuan.

Biasanya selang (I_{min} , I_{max}) sering digeser untuk alasan-alasan praktis menjadi selang $[0,L]$, yang dalam hal ini nilai intensitas 0 menyatakan hitam, nilai intensitas L menyatakan putih, sedangkan nilai intensitas antara 0 sampai L bergeser dari hitam ke putih. Sebagai contoh citra *grayscale* dengan 256 level artinya mempunyai skala abu dari 0 sampai 255 atau $[0,255]$, yang dalam hal ini intensitas 0 menyatakan hitam, intensitas 255 menyatakan putih, dan nilai antara 0 sampai 255 menyatakan warna keabuan yang terletak antara hitam dan putih.



Gambar 2.4 Citra hasil operasi *grayscale*

2.2 Image Processing

2.2.1 Optical Flow

2.2.1.1 Gambaran Optical Flow

Optical Flow adalah perkiraan gerakan suatu bagian dari sebuah citra berdasarkan turunan intensitas cahayanya pada sebuah sekuen citra. Pada ruang 2D hal ini berarti seberapa jauh suatu piksel citra berpindah diantara dua *frame* citra yang berurutan. Sedangkan pada ruang 3D hal ini berarti seberapa jauh suatu *volume* pixel (*voxel*) berpindah pada dua *volume* yang berurutan. Perhitungan turunan dilakukan berdasarkan perubahan intensitas cahaya pada kedua *frame* citra maupun *volume*. Perubahan intensitas cahaya pada suatu bagian citra dapat disebabkan oleh gerakan yang dilakukan oleh obyek, gerakan sumber cahaya, ataupun perubahan sudut pandang.

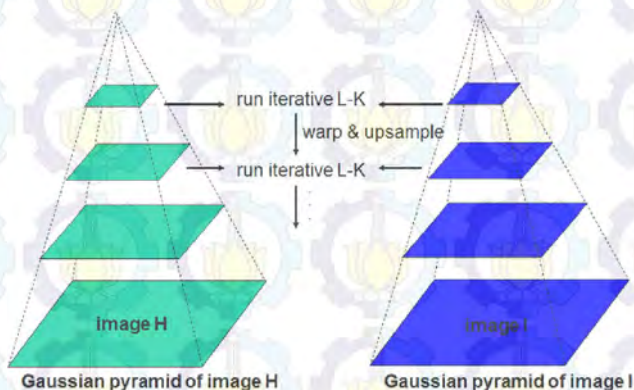
2.2.1.2 Algoritma Lucas-Kanade

Algoritma *Lucas-Kanade*, pertama kali diajukan pada tahun 1981. Pada awalnya algoritma ini adalah sebuah usaha untuk mencari teknik registrasi citra yang cepat dengan memanfaatkan *gradient* intensitas spasial. Pada perkembangannya, algoritma ini kemudian menjadi salah satu algoritma *optical flow* yang penting. Berbeda dengan algoritma *Horn-Schunk* yang bekerja berbasis pada keseluruhan citra, algoritma ini bekerja berdasar pada informasi lokal yang diturunkan dari *window* kecil (*patch*) disekeliling titik yang diperhitungkan dalam proses. Kelemahan digunakan *window* local kecil pada algoritma *Lucas-Kanade* adalah tidak terdeteksinya gerakan-gerakan yang besar karena gerakan-gerakan tersebut jatuh diluar *window*. Akibatnya terjadi *miss* pada gerakan yang besar. Permasalahan ini kemudian dapat diatasi dengan mengimplementasikan penyelesaian dengan prinsip piramida, yaitu *pyramidal Lucas-Kanade*. Prinsip ini merupakan penyelesaian berdasar iterasi dari level detail citra paling rendah hingga *level* detail citra paling tinggi.

2.2.1.3 Pyramidal Lukas-Kanade

Penyelesaian algoritma *Lukas-Kanade* dengan pendekatan piramida, atau yang umum disebut dengan *Pyramidal Lukas-Kanade*

diajukan pertama kali oleh Jean-Yves (Bouguet, pada tahun 2000). Pendekatan ini menggunakan prinsip piramida, yaitu bekerja mulai dari detil citra paling rendah hingga detil citra paling tinggi yang diilustrasikan pada gambar 2.5. Tujuannya adalah agar gerakan-gerakan yang “besar” dapat ikut diperhitungkan. Sementara asumsi yang digunakan pada algoritma *Lucas-Kanade* adalah gerakan yang “kecil” dan koheren, sehingga tidak dapat menangkap gerakan yang “besar”. Solusi untuk dapat menangkap gerakan yang “besar” pada algoritma *Lucas-Kanade* adalah dengan menggunakan *window* yang besar. Tetapi, akibat dari penggunaan *window* yang besar sering kali membuat gerakan yang ditangkap adalah gerakan yang tidak koheren. Algoritma *Pyramidal Lukas-Kanade* dapat menyelesaikan permasalahan tersebut tanpa menghilangkan asumsi gerakan-gerakan yang koheren.



Gambar 2.5 *Pyramidal Optical Flow*

Algoritma *Pyramidal Lucas-Kanade* pertama bekerja pada layer piramida paling tinggi. Kemudian hasilnya digunakan sebagai titik awal untuk bekerja pada *layer* dibawahnya. Hal ini berlanjut hinggamencapai *level* paling rendah. [6]

2.2.2 *Template Matching*

Template matching adalah sebuah teknik dalam pengolahan citra digital untuk menemukan bagian-bagian kecil dari gambar yang cocok

dengan template gambar. Energi cahaya yang terpancar dari suatu bentuk mengenai retina mata dan diubah menjadi energi neural yang kemudian dikirim ke otak. Selanjutnya terjadi pencarian di antara template - template yang ada. Jika sebuah template ditemukan sesuai (*match*) dengan pola tadi, maka subjek dapat mengenal bentuk tersebut. Setelah kecocokan antara objek dan *template* terjadi, proses lebih lanjut dan interpretasi terhadap objek bisa terjadi. [7]

2.2.3 *Human Detection*

Dalam proses pengenalan target, untuk mengetahui ada mengetahui apa saja yang ada di depan *user* akan sangat membantu *user* dalam berinteraksi. *Human Detection* dipilih sebagai salah satu fitur yang diberikan untuk membantu *user*. *Human Detection* adalah sebuah metode untuk mendeteksi ada atau tidaknya orang yang berada di depan *user*. Metode ini memanfaatkan library tambahan yang ada pada *emgucv* yaitu “*haarcascade_upperbody.xml*”. Metode ini mendeteksi bagian atas dari manusia yang nantinya akan diproses dan dikomunikasikan kepada *user* apabila terdeteksi manusia yang berada di depannya.

2.2.4. *Emgu CV*

EmguCV adalah wrapper .Net untuk *OpenCV*. Dengan *EmguCV*, fungsi-fungsi dalam *OpenCV* bisa dipanggil melalui bahasa pemrograman yang *compatible* dengan .NET seperti C#, VB, dan VC++. Keuntungan menggunakan *EmguCV* yang paling utama adalah *library* ini sepenuhnya ditulis dengan C# yang mana tentunya lebih *safe* karena pembuatan *object* atau pun *reference* di-manage oleh *garbage collector*. Selain itu, *Emgu CV* juga *cross platform* sehingga dapat di-compile lewat Mono dan dijalankan di atas sistem operasi Linux atau Mac OS.



Gambar 2.6 Logo *EmguCV*

2.2.5. Open CV

OpenCV (Open source Computer Vision) merupakan sebuah library dari fungsi programming untuk real-time computer vision. OpenCV dapat dimanfaatkan oleh program-program lainnya (seperti C++, C dan Phyton) untuk melakukan pengambilan, pengolahan serta penampilan data gambar, baik dalam bentuk image dan video maupun real-time video. OpenCV bersifat Open Source (dapat digunakan secara bebas) baik untuk akademik maupun untuk komersil, yang dapat berjalan dengan operating system Windows, Linus, Android dan Mac.



Gambar 2.7 Logo OpenCV[8]

2.3 Jenis Senjata

2.3.1 *Machine Gun*

Machine Gun (senapan mesin) adalah senjata api otomatis dipasang pada titik atau dibawa secara *portable* oleh setiap orang. Dirancang untuk menembak peluru secara berurutan dari sabuk amunisi atau magasin dan memiliki tingkat kecepatan menembak pada 3-1800 putaran per menit dan memiliki jarak efektif yang jauh bergantung dari kaliber yang digunakan.



Gambar 2.8 *M249 Light Machine Gun*

2.3.2 Submachine Gun

Submachine Gun adalah sebuah senjata api yang menggabungkan kemampuan menembak otomatis senapan mesin dengan amunisi pistol. Konsep senjata api seperti ini pertama kali dicoba pada tahun 1900-an, yaitu pistol yang diberi popor dan menembak secara otomatis.



Gambar 2.9 *MP5 Submachine Gun*

2.3.3 Assault Rifle

Assault Rifle (Senapan serbu) adalah senjata api otomatis yang merupakan senapan laras panjang atau karabin, yang memiliki pilihan tembakan (*selective-fire*), dan menggunakan amunisi kaliber menengah. Senapan serbu masuk dalam kategori di antara senapan mesin ringan, yang berfungsi untuk menembak secara *full*-otomatis sebagai senjata pendukung, dan *submachine gun*, senjata otomatis yang menggunakan peluru ukuran peluru pistol sebagai senjata api jarak dekat. Senapan

serbu sudah menjadi persenjataan standar untuk tentara modern, menggantikan senapan laras panjang era Perang Dunia II.



Gambar 2.10 M4 Assault Rifle

2.4 Data Acquisition Module

Data Acquisition Module adalah suatu *Module* yang memberikan hasil akuisisi data menuju computer. Sedangkan *Data Acquisition* adalah proses pengambilan sampel dari sebuah sinyal yang mengukur kondisi fisik dunia nyata dan mengkonversi sampel yang dihasilkan menjadi nilai numerik digital sehingga dapat diproses oleh komputer. Sistem akuisisi data biasanya mengkonversi bentuk gelombang analog menjadi nilai digital untuk diproses. Komponen sistem akuisisi data meliputi:

- Sensor yang mengkonversi parameter fisik menjadi sinyal-sinyal listrik.
- Sinyal pengkondisi rangkaian untuk mengkonversi sinyal sensor menjadi bentuk yang dapat dikonversi ke nilai digital.
- *Analog-to-digital converter*, yang mengkonversi sinyal sensor yang berupa analog menuju sinyal digital.



Gambar 2.11 Advantech Portable Data Acquisition Module [9]

2.5 Pulse Width Modulation

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu periode, untuk mendapatkan tegangan rata-rata yang berbeda. Beberapa contoh aplikasi PWM adalah pemodulasian data untuk telekomunikasi, pengontrolan daya atau tegangan yang masuk ke beban, regulator tegangan, audio effect dan penguatan, serta aplikasi-aplikasi lainnya. Aplikasi PWM berbasis mikrokontroler biasanya berupa pengendalian kecepatan motor DC, pengendalian motor servo, pengaturan nyala terang LED dan lain sebagainya.

Sinyal PWM pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar Pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. Artinya, Sinyal PWM memiliki frekuensi gelombang yang tetap namun duty cycle bervariasi (antara 0% hingga 100%).[9]

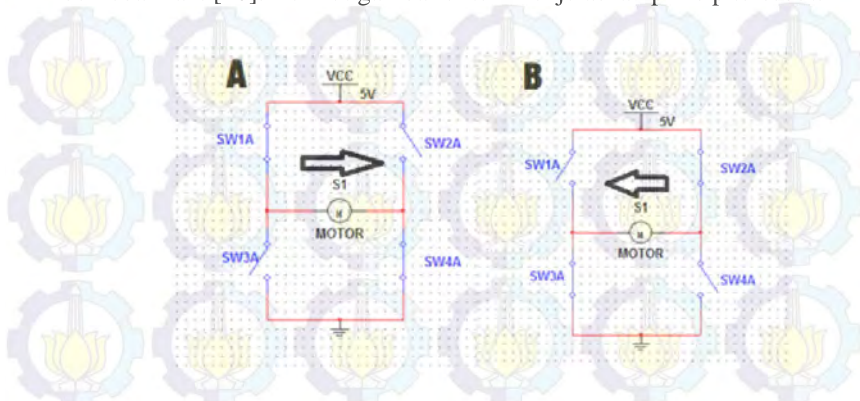


Gambar 2.12 Presentase *Duty Cycle* pada PWM

2.6 Driver Motor

Driver motor yang biasa digunakan adalah konfigurasi *H-Bridge* dimana arah pergerakan motor dapat dikendalikan sesuai dengan arah arum jam atau berlawanan dengan jarum jam. Pada prinsipnya, rangkaian ini terdiri dari empat buah saklar yang disusun sedemikian

rupa sehingga motor DC dapat dialiri arus dengan arah yang berkebalikan. [10]. Berikut gambar untuk menjelaskan prinsip tersebut:

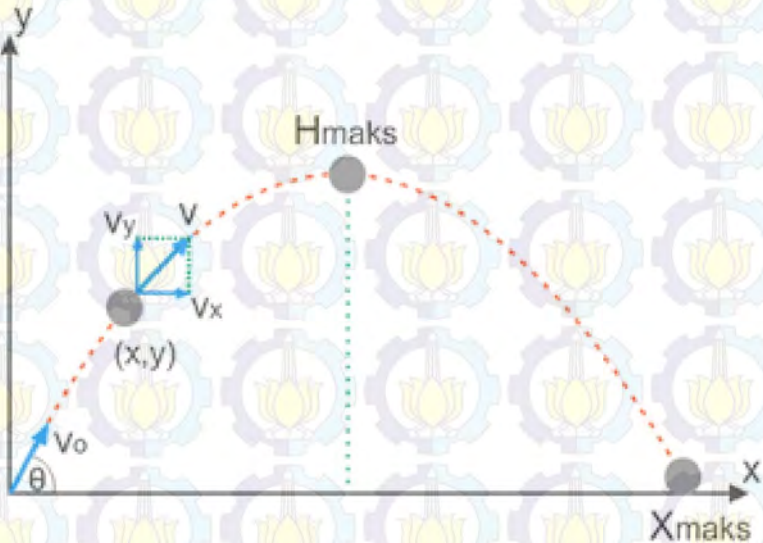


Gambar 2.13 Konfigurasi rangkaian *H-Bridge*

Prinsip kerja dari rangkaian H-bridge adalah mengatur arah alir arus pada motor dengan cara membuka atau menutup switch yang ada sehingga motor dapat diatur untuk bergerak searah jarum jam atau berlawanan dengan jarum jam. Ketika SW1A dan SW4A tertutup sedangkan SW2A dan SW3A terbuka (seperti pada gambar 2.11 A) maka sisi kiri dari motor akan tersambung dengan VCC dan sisi sebelah kanan motor akan terhubung dengan ground sehingga kondisi ini akan membuat arah pergerakan motor kekanan atau searah dengan jarum jam. Begitupun juga dengan kondisi sebaliknya, pada saat SW2A dan SW3A tertutup sedangkan SW1A dan SW4A terbuka (seperti pada gambar 2.11 B), maka arah alir arus akan bergerak dari sisi kanan motor menuju ke sisi kiri motor karena sisi kanan motor terhubung dengan VCC sedangkan sisi kiri motor terhubung dengan ground sehingga motor akan bergerak kearah kiri atau berlawanan dengan jarum jam. Pada penerapannya, switch ini biasanya diganti dengan saklar otomatis, salah satunya adalah dengan komponen transistor dimana untuk mengalirkan arus dari sisi kolektor kesisi emitor (*short*), kita hanya perlu memberikan arus forward pada basis, sehingga fungsi transistor tersebut akan berperan sebagai saklar.

2.7 Gerak Peluru

Gerak peluru atau parabola adalah perpaduan antara gerak horizontal (searah dengan sumbu x) dengan vertikal (searah sumbu y). Pada gerak horizontal bersifat GLB (Gerak Lurus Beraturan) karena gesekan udara diabaikan. Sedangkan pada gerak vertikal bersifat GLBB (Gerak Lurus Berubah Beraturan) karena pengaruh percepatan gravitasi bumi (g) terhadap kecepatan vertikal.



Gambar 2.14 Lintasan gerak peluru

Perumusan jarak maksimal gerak peluru (X_{maks}) dapat dihitung dengan:

$$X_{maks} = \frac{V_0^2 \cdot \sin^2 \alpha}{g} \quad (2-1)$$

Dari perumusan tersebut, gesekan udara dan pengaruh massa diabaikan. Maka kecepatan awal lekat benda mempengaruhi jarak maksimum tempuh benda.

2.8 Penelitian Terkait

2.8.1 CIWS (Close In Weapon System)

Close In Weapon System (CIWS), sering diucapkan *sea-whiz*, adalah senjata titik pertahanan point-defense untuk mendeteksi dan menghancurkan rudal jarak pendek yang masuk dan pesawat musuh yang telah menembus pertahanan luar, biasanya dipasang pada kapal dalam kapasitas angkatan laut.



Gambar 2.15 Phalanx CIWS[11]

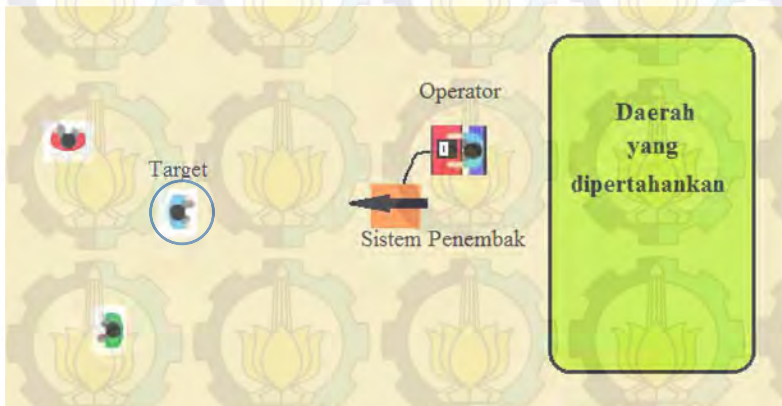


BAB III

PERANCANGAN SISTEM

Pada bab ini akan dibahas perancangan keseluruhan sistem dimulai dari penjelasan bagaimana skenario pelacakan dan penguncian target sampai pada perancangan *software* dan *hardware* dari sistem pelacakan dan penguncian target otomatis.

Pada tugas akhir ini memanfaatkan teknologi pengolah citra untuk melakukan pelacakan dan penguncian target secara otomatis. Sistem ini digunakan untuk mempertahankan suatu daerah dan berkonfigurasi pertahanan statis. Pada sistem ini *operator* memilih sasaran yang dirasa akan membahayakan dan sistem akan melacak dan mengunci sasaran yang telah dipilih oleh operator tersebut. Berikut adalah ilustrasi dan penjelasan peletakan sistem serta proses dari pemilihan sasaran, pelacakan, dan penguncian sasaran dilakukan.

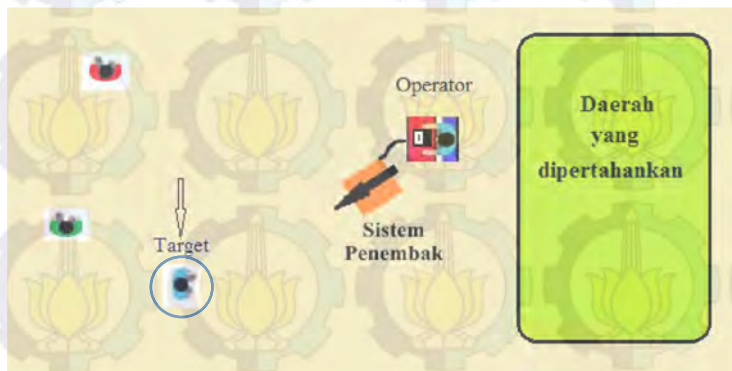


Gambar 3.1 Skenario pemilihan target

Sistem diletakkan untuk mempertahankan suatu *area* atau daerah yang dipertahankan. Sistem ini peletakannya statis sehingga disebut juga sebagai sistem pertahanan statis atau titik. Sasaran utama untuk sistem ini adalah manusia sehingga peletakan ketinggian sistem berada pada ketinggian 80cm hingga 150 cm. Sistem ini membutuhkan seorang

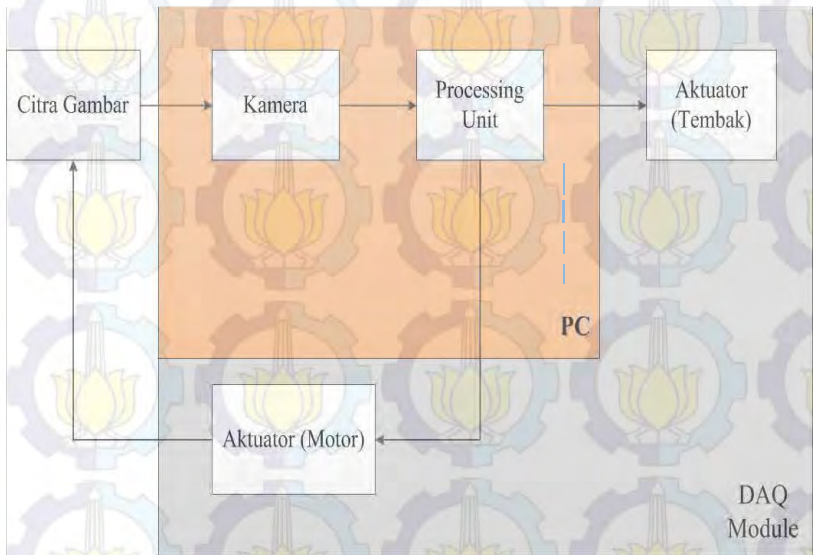
operator untuk mengoperasikannya. Saat dioperasikan, sistem akan menampilkan citra dan menampilkan keberadaan orang yang ada didepan sistem tersebut. Jika *operator* menganggap ada seseorang yang membahayakan, maka *operator* cukup melakukan pemilihan sasaran terhadap orang yang dicurigai lalu sistem akan mengikuti pergerakan sasaran yang telah dipilih. *Operator* dapat melakukan penggantian sasaran maupun menghapus sasaran yang telah dipilih. Jika sasaran dianggap perlu ditembak jatuh oleh *operator*, maka *operator* juga dapat memberikan perintah penembakan terhadap target melalui *processing unit* dan dapat memilih sasaran berikutnya jika diperlukan. Semua proses yang diperintahkan oleh *operator* dilakukan melalui *processing unit* atau komputer tanpa menyentuh sistem *hardware* (sistem penembak) secara langsung sehingga *operator* dapat berada pada jarak yang aman.

Pada gambar 3.1 diilustrasikan *operator* memilih orang berbaju biru sebagai sasaran atau target karena dianggap membahayakan atau mencurigakan. Sedangkan orang berbaju merah dan hijau dianggap tidak membahayakan sehingga tidak dimasukkan dalam sasaran oleh operator. Setelah pemilihan sasaran dilakukan, *processing unit* akan melacak pergerakan sasaran memerintahkan sistem penembak untuk mengikuti pergerakan dari sasaran tersebut. Berikut gambaran dari sistem penembak yang mengikuti pergerakan dari sasaran yang telah dipilih.

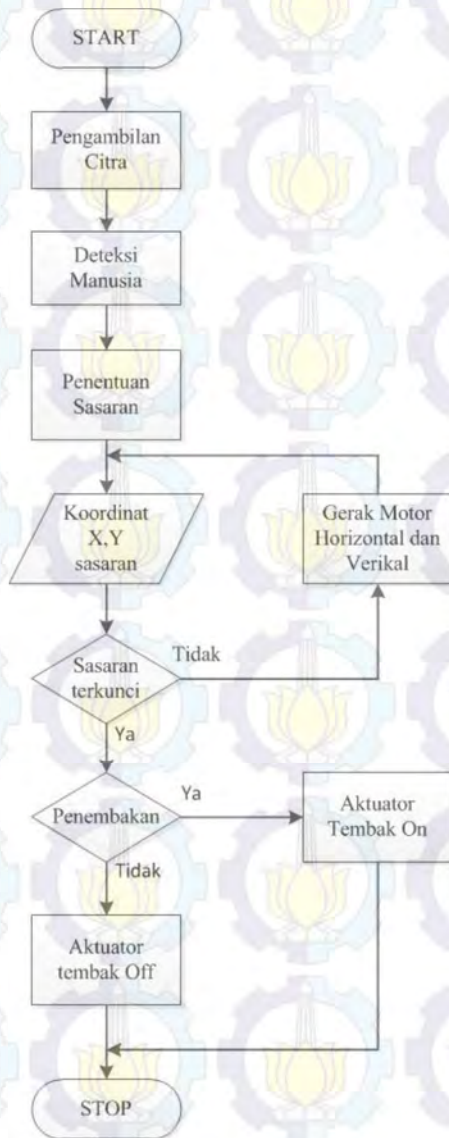


Gambar 3.2 Skenario sistem penembak mengikuti pergerakan target

Pada gambar 3.2 diilustrasikan sistem penembak akan tetap mengarah pada sasaran target berbaju biru yang pergerakannya tunjukkan oleh tanda panah. Sistem akan melacak dan mengunci sasaran yang telah dipilih ke manapun sasaran bergerak dalam *coverage area* sistem penembak. Terlihat pada gambar tersebut sistem penembak tetap mengarah menuju sasaran yang telah bergerak. Berikutnya jika oleh *operator* yang mengoperasikan sistem tersebut merasa target tersebut membahayakan dan perlu dilakukannya penembakan, maka *operator* dapat memerintahkan sistem penembak untuk menembak sasaran tersebut. Penembakan dapat dilakukan tanpa *operator* menyentuh sistem penembak secara langsung. *Operator* cukup memberikan perintah penembakan melalui *processing unit*. Berikut diagram blok sistem dan *flowchart* sistem secara keseluruhan.



Gambar 3.3 Diagram Blok Sistem



Gambar 3.4 Flowchart Sistem

3.1 Perancangan Software

Tahapan ini dirancang algoritma pemrograman untuk melakukan proses berupa pengambilan citra, *grayscale*, *optical flow*, pemilihan sasaran, pergerakan manual, serta penembakan. Semua proses perangkat lunak dilakukan oleh *processing unit* atau komputer. Algoritma pemrograman ini digunakan untuk mengatur hardware berupa DAQ module dan mengatur pergerakan motor untuk gerak vertikal, horizontal, dan menembak.



Gambar 3.5 Tampilan software

3.1.1 Pengambilan Citra

Data citra yang didapat pada input citra dari kamera disimpan dalam bentuk variabel frame. Variabel inilah yang akan diolah dengan metode *optical flow* input citra. Pada sistem ini data citra yang disimpan memiliki ukuran 640x480 piksel. Semakin besar ukuran piksel yang disimpan, maka semakin besar proses yang terjadi sehingga dapat menyebabkan *delay* pada sistem.



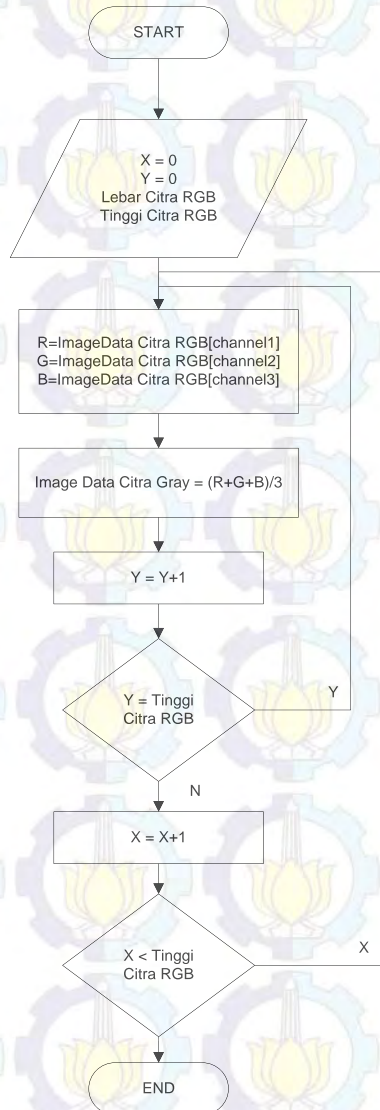
Gambar 3.6 Citra frame

3.1.2 *Grayscale*

Langkah pertama mengolah data citra untuk proses *optical flow* adalah dengan mengubah data citra awal yang bernilai RGB menjadi *grayscale*. Proses ini dilakukan karena *RGB* memiliki 24 bit, dimana masing-masing kanal warna memiliki nilai 8 bit yaitu 8 bit untuk *Red*, 8 bit untuk *Green*, dan 8 bit untuk *Blue*. Jika tidak dikonversi kedalam bentuk *grayscale*, maka perhitungan nilai piksel akan sangat sangat banyak sehingga memperlama proses. Maka perubahan ke *greyscale* membantu penghitungan untuk proses *optical flow* karena metode *surf* memerlukan frame yang bernilai 8 bit untuk mengurangi perhitungan yang kompleks.



Gambar 3.7 Video hasil *capture* dalam RGB diubah ke *Grayscale*



Gambar 3.8 Flowchart grayscale

Pada tugas akhir ini, digunakan fungsi yang sudah ada dalam library emgucv untuk mengubah dari citra RGB ke citra *grayscale*. berikut adalah prosedurnya :

```
Dim citraasli As New Image(Of Bgr, Byte)(640, 480)
```

```
frame = citraasli
```

```
grayB = frame.Convert(Of Emgu.CV.Structure.Gray, Byte)()
```

citraasli : merupakan sumber frame yang akan dikonversi.

frame : merupakan wadah pengganti sumber frame yang akan di konversi.

grayB : merupakan frame yang akan dipakai sebagai wadah hasil konversi.

.Convert(Of Emgu.CV.Structure.Gray, Byte)()

: Perintah untuk mengkonversi dari RGB ke GRAY

Secara Manual Mengubah RGB kedalam bentuk Grayscale dalam dilakukan dengan cara menjumlah data gambar tiap kanal R,G, dan B dan dibagi 3. Hal tersebut dapat dilihat pada gambar 3.7 *flowchart grayscale* diatas.

$$\text{Image data citra gray} = \frac{R+G+B}{3} \quad (3-1)$$

3.1.3 Optical Flow

Metode *optical flow* merupakan metode yang menggunakan intensitas cahaya sebagai dasar untuk mendeteksi objek. Memiliki keuntungan karena menghasilkan vector gerakan dengan tingkat kepadatan yang tinggi. Pada tugas akhir ini, sasaran dipilih oleh *operator* dan sistem melakukan pelacakan dengan *optical flow* dengan referensi intensitas cahaya awal sasaran yang dipilih tersebut.

Dalam *optical flow*, referensi yang telah dipilih akan dihitung intensitas *brightness* dari referensi tersebut. Dengan asumsi referensi piksel di lokasi (x,y,t) dengan intensitas $I(x,y,t)$ melakukan perpindahan sebanyak Δx , Δy , dan Δt pada frame berikutnya, ketetapan *brightness* dapat diberikan:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (3-2)$$

Dalam program yang dibuat, sebelum dilakukan pelacakan sasaran, pemilihan titik awal sasaran dilakukan terlebih dahulu oleh *operator* dan akan ditandai oleh program. Berikut tampilan penandaan sasaran oleh program:



Gambar 3.9 Penandaan sasaran oleh program

Dalam gambar 3.8 diatas, sasaran yang diberi tanda adalah tanda bundar biru muda. Program akan secara otomatis mengikuti tanda yang telah ditandai kemanapun bergerak dalam frame. Dalam program yang

telah dibuat, referensi yang digunakan untuk dilacak hanya 1 tanda saja. Tanda lain yang diberikan berikutnya akan diabaikan oleh program.

3.1.4 Haar cascade

Dalam tugas akhir ini digunakan metode haar cascade, yang merupakan bagian dari metode *viola jones*. Metode *viola jones* adalah metode pendeteksi objek yang terdapat EmguCV. Metode ini merupakan metode yang paling banyak digunakan untuk mendeteksi objek, hal ini dikarenakan metode *viola jones* memiliki algoritma yang efisien, sehingga tidak memerlukan waktu lama dalam melakukan proses pendeteksian objek. Teori ini menjumlahkan nilai-nilai dari beberapa citra dalam sebuah kotak *frame*. Penjumlahan tersebut dapat dilihat dalam formula di bawah ini,

$$\sum_k \delta_k B_k(I) \quad (3-3)$$

dimana, $\delta \in \{1, -1\}$ dan

$$B_k(I) = \sum_{i=u1(k)}^{u2(k)} \sum_{j=v1(k)}^{v2(k)} I_y \quad (3-4)$$

Beberapa fitur dapat dengan cepat di evaluasi oleh integral *image*, dimana formula integral *image* dapat di bawah ini,

$$I_y = \sum_{u=1}^i \sum_{v=1}^j I_{uv} \quad (3-5)$$

dimana u dan v adalah nilai dari pixel integral *image*. Maka dari itu penjumlahan nilai dari sebuah *frame* dapat dievaluasi dengan 4 tahap integral *image*. Maka dengan mudah kita dapat memeriksa bahwa

$$\sum_{i=u1}^{u2} \sum_{j=v1}^{v2} I_y = I_{u2v2} - I_{u1v2} - I_{u2v1} + I_{u1v1} \quad (3-6)$$

setiap fitur dari citra yang tertangkap dapat dievaluasi dengan metode integral *image*[12]. Algoritma deteksi *upper body* dapat dilihat pada tabel di bawah ini

1. Deteksi *upper body*

- Membuat sebuah detektor objek *cascade*.
- Mengambil *frame image* dari *image* yang diambil dan jalankan detektor.

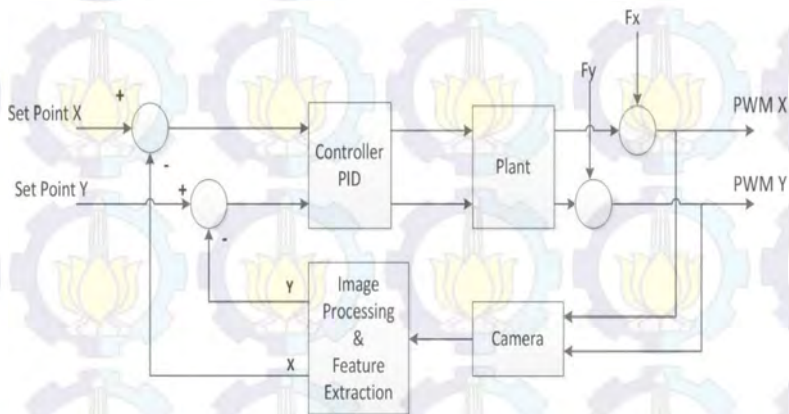
- Gambarkan sebuah kotak yang akan memberi tanda bila ada *upper body* yang terdeteksi.
- 2. Identifikasi fitur *upper body* untuk dilakukan pelacakan
 - Dapatkan informasi *upper body* dengan cara mengekstrak *Hue* dari setiap *frame image*.
 - Ubah citra tersebut ke dalam bentuk warna Hue, Saturation and Value (HSV).
 - Tampilkan *channel data Hue* dan gambar kotak di sekitar *upper body*.
 - Deteksi *upper body* di sekitar area *frame image*.
- 3. Pelacakan *upper body*
 - Buat sebuah pelacak objek.
 - Inisialisasi pelacak histogram menggunakan *channel pixel Hue*.
 - Buat sebuah *video* untuk menampilkan tiap *frame image*.
 - Lacak *upper body* di setiap *frame image* hingga pengambilan *frame* selesai
 - Beri sebuah kotak mengelilingi *upper body* pada objek yang terdeteksi
 - Tampilkan dalam *frame image* yang telah diolah

3.1.5 Kontrol PID

Pada *processing unit*, untuk mengatur agar gerakan motor dc yang digunakan penggerak vertikal dan horizontal pada sistem penembak diproses dengan control PID terlebih dahulu. Dalam metode pelacakan, selisih dari posisi piksel sasaran terhadap posisi posisi tengah frame (set point) untuk penguncian dianggap *error* untuk masukan E pada perhitungan kontrol PID oleh *processing unit* dan diperhitungkan seberapa besar kontrol pergerakan yang diperlukan untuk mengunci sasaran. Keluaran dari perhitungan ini adalah nilai arah gerak dan seberapa cepat motor perlu digerakkan.

Pada sistem ini yang digunakan adalah kontrol PID dengan bias dimana masukan bias (F_x untuk X dan F_y untuk Y) digunakan untuk mengatasi efek berat senjata ke motor horizontal maupun vertikal yang menyebabkan adanya zona mati dimana memerlukan *duty cycle* minimal

untuk menggerakkan motor (motor tidak langsung bergerak begitu *duty cycle* naik lebih dari 0%). Keluaran nilai PID pada sistem ini adalah antara 0 hingga 4 dimana akan direpresentasikan menjadi nilai tegangan analog yang akan keluar dari putput analog pada *DAQ Module* yang akan mengatur keluaran PWM pada rangkaian PWM. Dimana gambaran kontrol PID adalah sebagai berikut:



Gambar 3.10 Diagram blok sistem kontrol PID

3.1.6 Koneksi Advantech DAQ Module

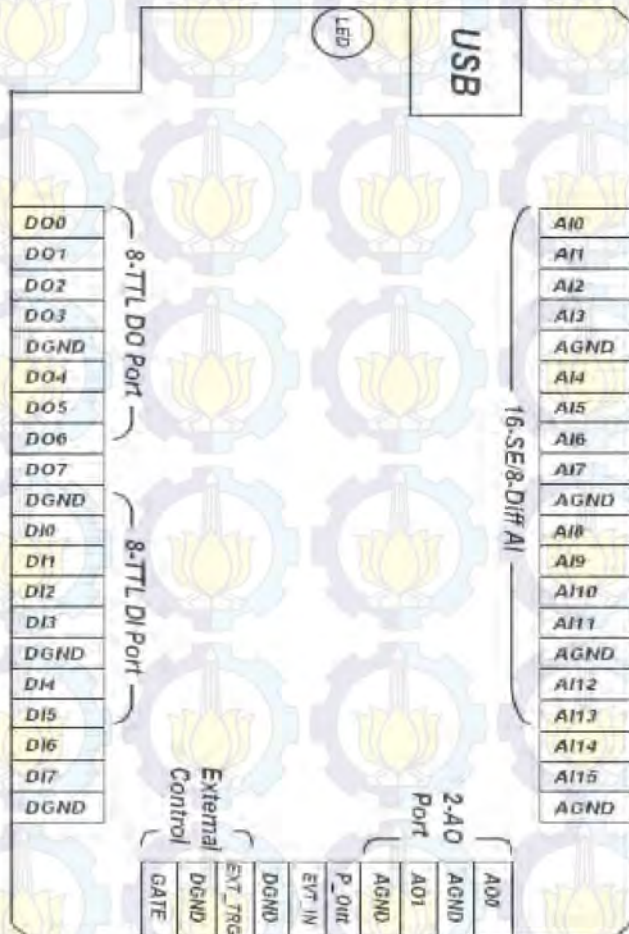
Pada tugas akhir ini untuk menghubungkan *processing unit* dengan *hardware* yang ada menggunakan *portable data acquisition module*. Untuk pembacaan *analog input*, *digital output*, dan *analog output* diperintah oleh *processing unit* via modul ini

3.2 Hardware

3.2.1 Advantech USB-4711A

Pada tugas akhir ini digunakan Advantech USB-4711A yang memiliki 8 port *digital input*, 8 port *digital output*, 2 port *analog output*,

16 port analog input, 4 port digital ground, 6 port analog ground, dan 3 port external control. Berikut konfigurasi port pada Advantech USB-4711A dan gambaran modulnya.



Gambar 3.11 Konfigurasi port Advantech USB-4711A



Gambar 3.12 Advantech USB-4711A

Pada tugas akhir ini digunakan 2 port *analog output* untuk pengatur *duty cycle* pada rangkaian pembangkit PWM (*Pulse Width Modulation*), 5 port *digital out* dimana 4 port untuk mengatur arah gerak motor (mengatur gerak vertikal dan horizontal) dan 1 port untuk pengatur penembakan, dan 1 port *analog input* untuk mengatur *emergency stop input* dari sensor *proximity*. Untuk lengkapnya penggunaan port pada Advantech USB-4711A dapat dilihat pada table berikut:

Tabel 3.1 Fungsi Penggunaan Port Advantech USB-4711

Port	Fungsi
DO0	Pin 1 Driver Motor Vertikal
DO1	Pin 2 Driver Motor Vertikal
DO2	Pin 1 Driver Motor Horizontal
DO3	Pin 2 Driver Motor Horizontal
DO4	Perintah menembak
AO0	Pengatur masukan Pembangkit PWM (Vertikal)
AO1	Pengatur masukan Pembangkit PWM (Horizontal)
AI0	Masukan Sensor <i>Proximity</i>
DGND	Referensi Ground
AGND	Referensi Ground

3.2.2 Sensor *Proximity*

Sensor *Proximity* disini digunakan untuk memberi batasan gerak pada gerakan horizontal (*emergency stop* pada sistem). Sensor *proximity*

yang digunakan adalah Autonics PR8-2DN. Sensor ini memiliki parameter sebagai berikut:

- Jarak deteksi : $2\text{mm} \pm 10\%$
- *Power Supply* : 12-24V
- Jenis Deteksi : Logam
- Output sensor : *Active Low*

Berikut gambaran sensor *Proximity* yang digunakan

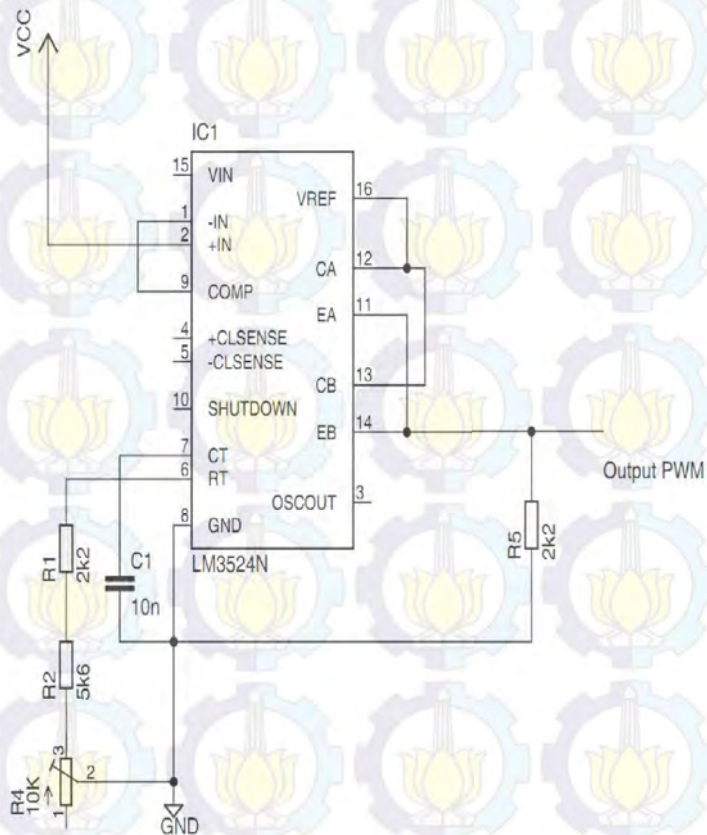


Gambar 3.13 Sensor *Proximity*

3.2.3 Rangkaian Pembangkit PWM (*Pulse Width Modulation*)

Pada blok ini menjelaskan rangkaian yang berfungsi untuk menghasilkan *pulse width modulation* yang akan masuk ke *driver* motor. Pada rangkaian ini dihasilkan *pulse width modulation* dari rentang *duty cycle* 0% hingga 100%. Rangkaian ini menggunakan IC LM3524 sebagai penghasil *pulse width modulation*. Referensi yang digunakan untuk mengatur *duty cycle* adalah masukan *analog input* dari Advantech

USB-4711A. Masukan untuk mengatur *duty cycle* adalah tegangan 0 volt hingga tegangan 5 volt. Berikut skematik rangkaian pembangkit *pulse width modulation*.



Gambar 3.14 Skematik rangkaian pembangkit PWM

Untuk menggerakkan motor horizontal dan vertikal diperlukan masing-masing 1 rangkaian, sehingga terdapat 2 buah rangkaian serupa pada rangkaian yang dibuat. Berikut gambaran dari rangkaian yang telah dibuat:



Gambar 3.15 Rangkaian pembangkit PWM (dalam kotak)

Prinsip kerja dari rangkaian tersebut adalah, semakin besar tegangan masukan pada port 2, maka semakin besar *duty cycle* yang dihasilkan pada port 7 dari IC 3524 tersebut. Tegangan keluaran yang diinginkan diambil pada port 11 dan 14. Untuk mengatur tegangan keluaran pada *duty cycle*, pada port 12 dan 13 diberikan tegangan masukan. Keluaran tegangan *duty cycle* mengikuti nilai tegangan yang memasuki port 12 dan 13 tersebut. Sedangkan untuk mengatur frekuensi keluaran pada keluaran, bergantung pada perbandingan resistansi dari resistor pada port 6 dan kapasitor pada port 7. Kedua rangkaian ini akan menghasilkan gelombang gerigi segitiga yang akan dikomparasikan dengan tegangan input dari port 2. Keluaran dari komparator tersebut mengatur waktu *on* dan *off* pada keluaran port 12 dan 13 sehingga keluaran akhir yang didapatkan adalah gelombang kotak dengan periode *on* (tegangan keluaran sama dengan tegangan masukan port 12 dan 13) dan *off* (tegangan keluaran 0volt) yang diinginkan. Perbandingan antara waktu *on* dan *off* itulah yang disebut dengan *duty cycle*. Berikut rumusan dari nilai *duty cycle*:

$$Duty\ Cycle = \frac{T_{on}}{T_{off} + T_{on}} \times 100\% \quad (3-7)$$

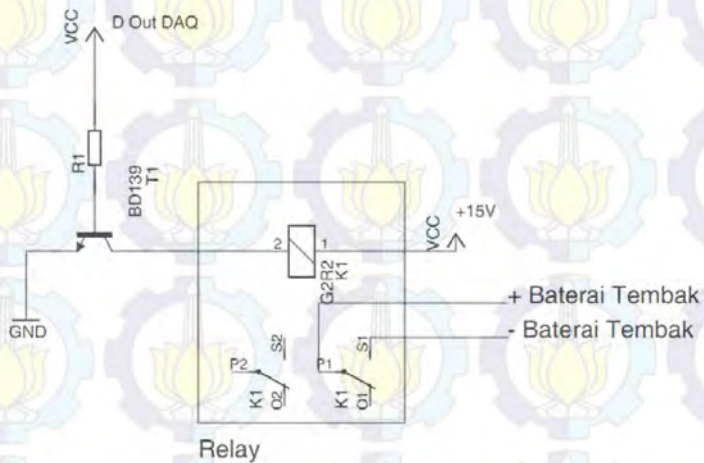
T_{on} : Waktu tegangan keluaran maksimal

T_{off} : Waktu tegangan keluaran 0 volt.

Keluaran dari rangkaian pembangkit PWM ini akan menuju ke *driver* motor untuk mengatur kecepatan motor. Keluaran rangkaian PWM vertikal akan menuju ke pin masukan PWM pada *driver* motor vertikal sedangkan keluaran rangkaian PWM horizontal akan menuju ke pin masukan PWM pada *driver* motor horizontal

3.2.4 Aktuator Tembak

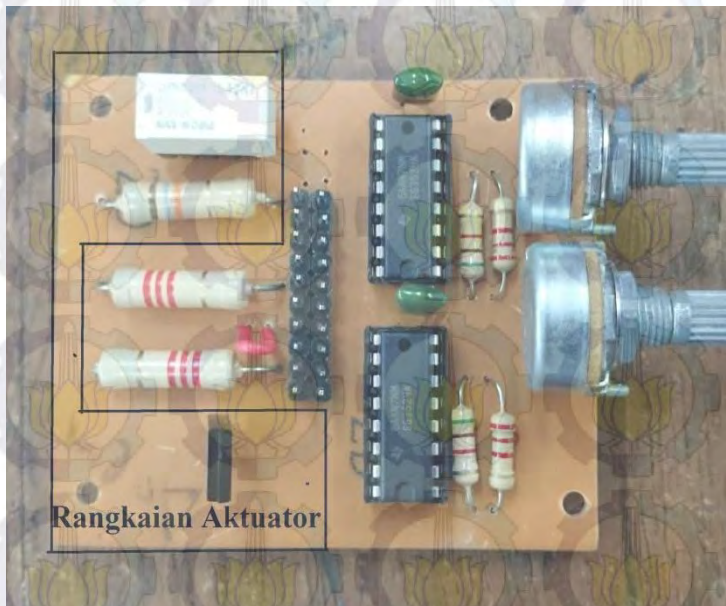
Disini dibuat sebuah rangkian untuk mengatur penembakan dari senjata yang digunakan. Berikut skematik dari rangkaian aktuator tembak yang telah dibuat:



Gambar 3.16 Skematik rangkaian aktuator tembak

Fungsi dari transistor BD139 pada rangkaian tersebut adalah untuk mengaktifkan relay. Saat basis pada transistor diberi tegangan, maka transistor akan mengalami saturasi dan membuat relay mengaktifkan

relay. Jika relay aktif, maka rangkaian yang sebelumnya *normally open* akan menjadi *close* sehingga rangkaian baterai tembak menjadi *close loop* dan membuat senjata yang dipasang menembak. Untuk menghentikan penembakan, maka cukup menghentikan tegangan basis menjadi 0. Jika tegangan basis 0volt, maka transistor BD139 menjadi *open circuit* pada emitter dan kolektornya sehingga membuat relay menjadi *off* dan rangkaian loop baterai senjata menjadi *open loop* dan menghentikan siklus menembak. Berikut rangkaian untuk aktuator menembak.



Gambar 3.17 Rangkaian aktuator menembak

3.2.5 Driver Motor

Driver Motor disini menggunakan BTN7970. Dalam tugas akhir ini menggunakan 2 buah *driver motor*. 1 untuk motor penggerak vertikal dan 1 lagi untuk motor penggerak horizontal. Berikut gambaran dari *driver motor* yang digunakan:



Gambar 3.18 Driver motor BTN7870B

Supply yang digunakan untuk *driver* motor menggunakan 24VDC 5A. Sedangkan masukan driver motor terdapat pin 3 yaitu masukan PWM untuk mengatur kecepatan motor, dan in 1 dan in 2 untuk mengatur arah gerakan motor. Untuk mengarahkan pergerakan pergerakan atas dan bawah pada motor penggerak vertikal dan pergerakan kiri dan kanan pada motor penggerak horizontal memiliki kondisi tertentu. Kondisi yang dimaksud dapat dilihat pada table berikut ini:

Tabel 3.2 Tabel kebenaran driver motor Vertikal

Driver Motor Vertikal		
In 1	In 2	Aksi pergerakan
0	0	Diam
0	1	Atas
1	0	Bawah
1	1	Diam

Tabel 3.3 Tabel kebenaran driver motor Horizontal

Driver Motor Horizontal		
In 1	In 2	Aksi pergerakan
0	0	Diam
0	1	Kanan
1	0	Kiri
1	1	Diam

3.2.6 Motor DC

Motor yang digunakan sistem penggerak horizontal dan vertikal pada tugas akhir ini adalah motor DC yang bisa bergerak pada 2 arah yang berlawanan. Arah pergerakan motor ditentukan oleh pemberian tegangan DC pada kedua input motor. Berikut gambaran motor DC yang digunakan pada tugas akhir ini beserta peletakannya.



Gambar 3.19 Motor DC penggerak vertikal



Gambar 3.20 Motor DC penggerak horizontal

3.2.7 Senjata

Senjata yang digunakan pada tugas akhir ini adalah jenis elektrik *air soft gun*. Terdapat 2 buah senjata yang digunakan yaitu purwarupa M4 dengan perbandingan ukuran 1:1 dan perbandingan berat 1:1 dan purwarupa SIG552 dengan ukuran 1:2 dan perbandingan berat 1:15. Berikut gambaran dari senjata yang digunakan:



Gambar 3.21 Purwarupa M4 1:1



Gambar 3.22 Purwarupa SIG552 1:2

3.2.8 Kamera

Kamera yang digunakan pada tugas akhir ini adalah Logitech C920. Berikut gambaran kamera yang digunakan:



Gambar 3.23 Logitech C920

Berikut spesifikasi dari Logitech C920:

- Perekaman video hingga 1080p (1920 x 1080 piksel)
- Kompresi video H.264
- Lensa Carl Zeiss dengan 20 langkah *autofocus*
- Built-in dual stereo mics* dengan koreksi noise otomatis
- Koreksi otomatis pada kondisi cahaya yang rendah
- Hi-Speed USB 2.0 (USB 3.0 ready)*
- Tripod ready clip*
- 15 *megapixels* untuk *capture* foto.

3.2.9 Komputer

Pada tugas akhir ini digunakan sebuah komputer sebagai *processing unit*. Spesifikasi komputer yang digunakan pada tugas akhir ini adalah:

- | | |
|----------------------|---------------------------------------|
| • <i>Manufacture</i> | : Dell |
| • Model | : XPS L502X |
| • CPU | : Intel ® Core™ i7-2630QM
@2.00GHz |
| • RAM | : 4GB |
| • <i>System Type</i> | : 64-bit |

3.2.10 Sistem Penembak

Dalam bab ini dijelaskan perancangan keseluruhan sistem penembak. Berikut gambaran dari sistem penembak dan penjelasan penempatan *hardware* yang digunakan pada sistem meliputi *driver motor*, rangaian pembangkit PWM, senjata, sensor *proximity*, kamera, *DAQ module* dan motor DC untuk kedua jenis senjata.



Gambar 3.24 Rancangan sistem penembak dengan purwarupa SIG552



Gambar 3.25 Rancangan sistem penembak dengan purwarupa M4

Berikut penjelasan penempatan hardware:

No:

- 1 = Senjata
- 2 = Kamera
- 3 = Motor DC penggerak vertikal
- 4 = Sensor *Proximity*
- 5 = Rangkaian pembangkit PWM
- 6 = Driver motor DC
- 7 = Rangkaian aktuator penembak
- 8 = Advantech USB-4711A

Untuk peletakan motor penggerak horizontal ada di dalam bodi sistem. Berikut gambaran posisi motor DC penggerak horizontal:



Gambar 3.26 Posisi motor DC penggerak horizontal

BAB IV

PENGUJIAN

Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang. Bab ini bertujuan untuk mengetahui apakah tujuan dalam perancangan sistem pada tugas akhir ini telah terlaksana atau tidak. Pengujian pada bab ini terdiri dari pengujian keluaran rangkaian pembangkit PWM, serta pelacakan dan penguncian terhadap sasaran dengan dua senjata yang berbeda.

4.1 Pengujian Rangkaian Pembangkit PWM

Pengujian yang dilakukan pada bagian ini adalah pengujian untuk mengambil atau menyimpan data gelombang PWM yang dihasilkan oleh rangkaian pembangkit PWM. Pengujian dilakukan dengan memberi tegangan referensi antara 0 hingga 4 Volt dimana akan dikonversikan menuju presentase *duty cycle*. Semakin besar nilai tegangan referensi yang diberikan, maka *duty cycle* keluaran dari rangkaian akan semakin besar. Berikut gambaran dari pengujian yang dilakukan:

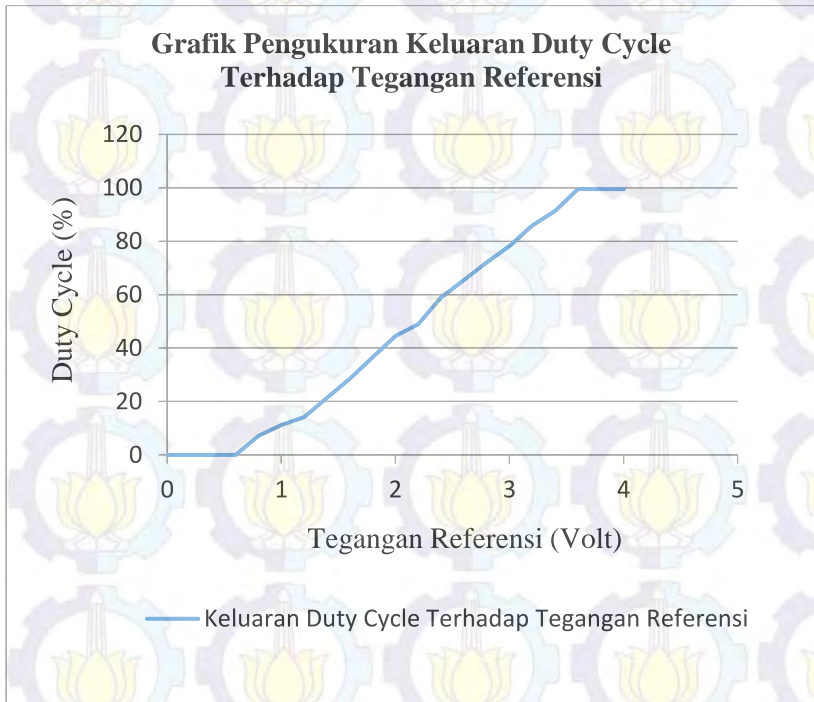


Gambar 4.1 Pengujian rangkaian pembangkit PWM

Pada pengujian ini dilakukan pengambilan data berupa keluaran *duty cycle* dari rangkaian pembangkit PWM (*Pulse Width Modulation*) tersebut setiap kenaikan 0,2volt pada tegangan referensi. Dari pengujian tersebut dapat diambil data-data sebagai berikut:

Tabel 4.1 Pengukuran keluaran *Duty Cycle* terhadap tegangan referensi

Tegangan referensi (Volt)	Duty Cycle (%)
0	0
0,2	0
0,4	0
0,6	0
0,8	7,07
1	11,17
1,2	14,14
1,4	21,32
1,6	28,43
1,8	36,45
2	44,44
2,2	48,98
2,4	58,88
2,6	65,48
2,8	72,08
3	78,17
3,2	85,79
3,4	91,37
3,6	99,49
3,8	99,49
4	99,49



Gambar 4.2 Grafik hasil pengukuran keluaran *duty cycle* terhadap tegangan referensi

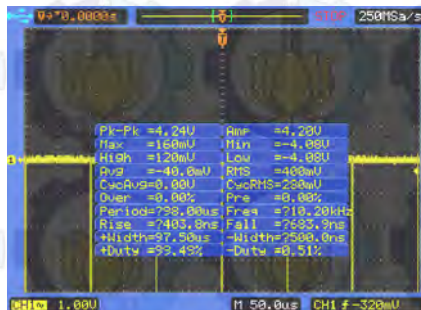
Dari data pengujian tersebut dapat diketahui bahwa keluaran *duty cycle* pada rangkaian pembangkit PWM yang telah dibuat dimulai pada saat tegangan referensi bernilai kurang dari 0,8 volt dan diatas 0,6volt serta mencapai nilai maksimum pada saat tegangan referensi bernilai 3,6volt. Dari data tersebut juga dapat diketahui bahwa setiap kenaikan tegangan sebanyak 0,1 volt terjadi kenaikan nilai *duty cycle* sebesar kurang lebih 3%. Untuk besaran tegangan *peak* keluaran pada rangkaian tersebut antara 0 hingga 4,56 volt Berikut beberapa gambar hasil pengukuran pada osiloskop:



Gambar 4.3 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 0,8Volt



Gambar 4.4 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 2,2Volt



Gambar 4.5 Keluaran rangkaian pembangkit PWM dengan tegangan referensi 3,6Volt

4.2 Pengujian Motor

Seperti dijelaskan pada bab sebelumnya pada tugas akhir ini digunakan 2 jenis purwarupa senjata dengan berat yang berbeda. Pengujian ini dilakukan untuk mengetahui nilai minimal *duty cycle* yang dibutuhkan untuk menggerakkan motor dengan beban yang berbeda. Untuk motor vertikal ditambahkan parameter berupa pengujian gerak dari sudut maksimum. Dari pengujian didapat data-data sebagai berikut:

Tabel 4.2 Pengujian motor DC vertikal

Kondisi beban	Massa Beban	Duty Cycle Minimum	
		Sudut 0	Sudut Maksimum
tanpa beban	0Kg	25%	25%
SIG 552	0,5Kg	32%	65%
M4	4Kg	32%	-

Dari pengujian dapat dianalisis yaitu pada saat sistem diberikan beban, maka dibutuhkan *duty cycle* minimum yang lebih besar untuk menggerakkan motor. Hal ini dapat dilihat dari saat kondisi tanpa beban dan dengan beban dimana dari pengujian terlihat kondisi tanpa beban membutuhkan *duty cycle* yang lebih kecil untuk menggerakkan motor dan kondisi dengan beban membutuhkan *duty cycle* yang lebih besar untuk menggerakkan motor.

Sedangkan sudut memiliki pengaruh besar untuk menggerakkan motor. Saat sudut maksimum (saat senjata menunduk atau mendongak), dibutuhkan *duty cycle* yang lebih besar untuk menggerakkan motor dapat dilihat dari kondisi tanpa beban dan dengan beban SIG 552 dimana membutuhkan *duty cycle* yang lebih besar untuk menggerakkan motor daripada pada saat sudut 0. Untuk purwarupa M4 yang memiliki massa 4kg, beban yang diberikan melebihi kapasitas motor sehingga pada sudut maksimum diberikan *duty cycle* maksimal tetap tidak bergerak. Sudut maksimum untuk purwarupa SIG552 adalah 15° dan untuk 7° untuk purwarupa M4.

Tabel 4.3 Pengujian motor DC horizontal

Kondisi Beban	Masssa Senjata	Duty Cycle Minimum
Tanpa Beban	0Kg	30%
SIG552	0,5Kg	35%
M4	4Kg	40%

Dari pengujian dapat dianalisis yaitu saat diberikan beban, maka dibutuhkan *duty cycle* minimum yang lebih besar untuk menggerakkan motor. Hal ini dapat dilihat dari saat kondisi tanpa beban dan dengan beban dimana dari pengujian terlihat kondisi tanpa beban membutuhkan *duty cycle* yang lebih kecil untuk menggerakkan motor.

Untuk motor horizontal terjadi kondisi dimana motor terdapat hambatan untuk bergerak sehingga terjadi keterlambatan bergerak. Hal ini disebabkan oleh mekanik pada motor horizontal.

4.3 Pengujian Pendeteksi Manusia

Dengan metode yang telah dijelaskan sebelumnya, dilakukan pengujian apakah sistem dapat mendeteksi orang pada hasil citra yang ditangkap. Pengujian dilakukan dengan metode *rough search* dimana sistem akan mendeteksi setiap orang yang dideteksi. Pengujian dilakukan pada siang hari dengan pencahayaan matahari dan tidak berawan

- Waktu : Pukul 15.30 WIB
Tempat : Lapangan Parkir Dosen Teknik Elektro ITS
Ketinggian : 90cm

Pengujian dilakukan dengan jumlah orang yang dideteksi bertambah. Dimulai dengan 1 orang, 2 orang, dan terakhir 3 orang. Manusia yang dideteksi dalam kondisi menghadap atau membelakangi

kamera yang digunakan pada sistem. Dari pengujian tersebut didapat data-data sebagai berikut

Tabel 4.4 Tabel pengujian deteksi manusia

Jumlah Orang	Jumlah Terdeteksi
1	1
2	1
3	2



Gambar 4.6 Pengujian pendeteksi manusia



Gambar 4.7 Pengujian deteksi manusia



Gambar 4.8 Pengujian deteksi manusia



Gambar 4.9 Pengujian deteksi manusia

Dari data diatas dapat disimpulkan bahwa untuk pendeteksian manusia pada pengujian ini terdapat kesalahan deteksi berupa *false positive* yaitu dimana benda bukan manusia terdeteksi sebagai manusia pada gambar 4.8 dan terdapat kesalahan berupa objek manusia tidak terdeteksi pada gambar 4.6, 4.7, dan 4.8. Dari data tersebut dapat dibuktikan juga arah menghadap objek mempengaruhi deteksi. Arah menghadap yang optimal adalah objek manusia menghadap kamera secara langsung atau membelakangi kamera jika ada kemiringan mempengaruhi pendeteksian.

4.4 Pengujian Pelacakan dan Penguncian Sasaran pada Siang Hari

Dengan menggunakan metode yang telah dijelaskan pada bagian perancangan sistem, maka dilakukan pengujian terhadap sistem yang telah dibuat. Pada pengujian kali ini dilakukan dengan purwarupa

purwarupa SIG 552 Pengujian awal dilakukan pada siang hari dengan cuaca berawan dan kondisi pencahayaan yang cukup.

- Waktu : Pukul 14.30 WIB
- Tempat : Lapangan Parkir Dosen Teknik Elektro ITS
- Ketinggian : 90cm

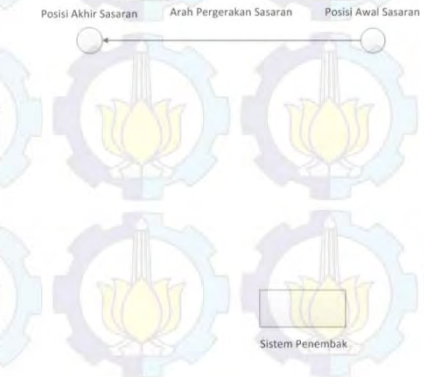
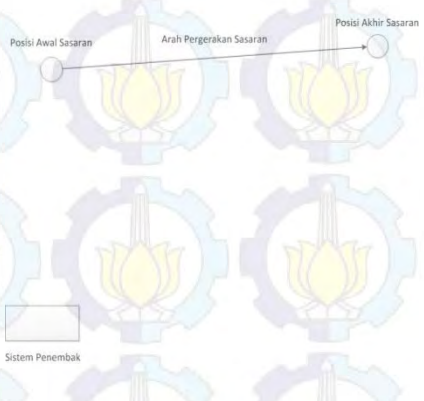
Pengujian dilakukan seberapa lama sistem dapat melacak dan mengunci sasaran dengan warna sasaran yang dirubah. Dari pengujian tersebut didapat data-data sebagai berikut:

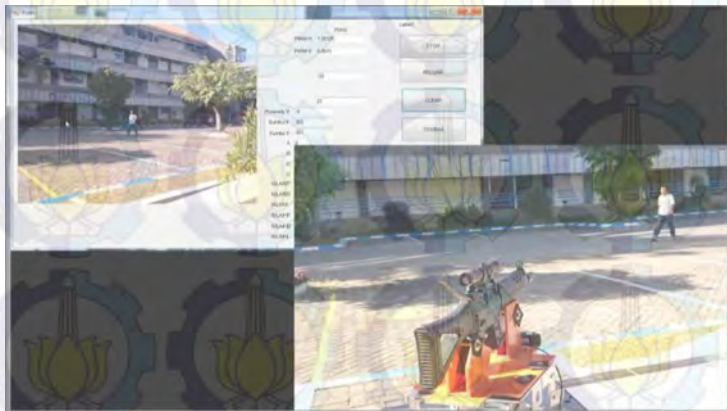
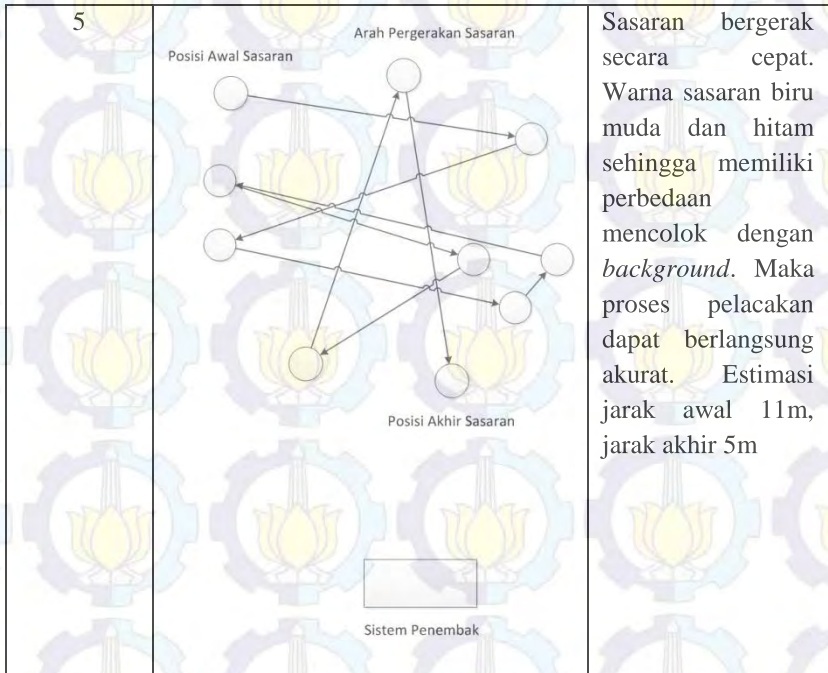
Tabel 4.5 Tabel pengujian pelacakan pada siang hari

Percobaan	Lama Pelacakan (detik)	Warna Sasaran
1	11	Biru
2	12	Biru
3	9	Merah
4	28	Merah
5	40	Biru Hitam

Tabel 4.6 Tabel alur gerak sasaran dan keterangan pengujian pelacakan pada siang hari

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna belakangnya sehingga sistem lepas dari sasaran. Estimasi jarak 7m akhir, 10m awal.</p>

2		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna belakangnya sehingga sistem lepas dari sasaran. Estimasi jarak awal 7m dan akhir 8m</p>
3		<p>Sasaran berwarna merah sehingga berbeda dengan warna <i>background</i>. Sasaran bergerak cepat sehingga sasaran terlepas dari sistem. Estimasi jarak 7m awal, 9m akhir.</p>
4		<p>Sasaran bergerak secara lambat. Warna sasaran merah sehingga berbeda dengan <i>background</i>. Maka proses pelacakan dapat berlangsung akurat. Estimasi jarak awal 7m, akhir 12m</p>



Gambar 4.10 Proses pengujian pelacakan pada siang hari

Selain itu dilakukan pengujian pelacakan kedua pada hari yang berbeda namun dengan posisi dudukan sistem senjata yang sama. Pengujian dilakukan pada kondisi cuaca cerah dan pencahayaan yang cukup. Pengujian dilakukan pada:

- Waktu : Pukul 13.30 WIB
Tempat : Lapangan Parkir Dosen Teknik Elektro ITS
Ketinggian : 90cm

Pengujian dilakukan seberapa lama sistem dapat melacak dan mengunci sasaran dengan 2 buah purwarupa senjata berbeda yang telah ada sebelumnya. Pada pengujian kali ini dilakukan dengan 2 jenis purwarupa yang berbeda yaitu purwarupa SIG 552 dengan massa purwarupa 0,5kg dan purwarupa M4 dengan massa purwarupa 4kg. Pada pengujian ini sasaran yang digunakan menggunakan warna yang tetap yaitu warna putih. Dari pengujian tersebut didapat data-data sebagai berikut:

Tabel 4.7 Tabel pengujian pelacakan kedua pada siang hari purwarupa SIG552

Percobaan	Lama Pelacakan (detik)
1	42
2	61

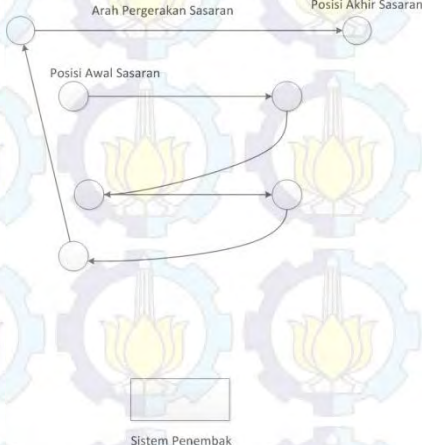
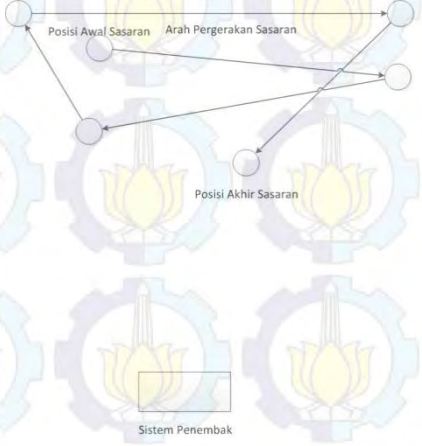
Tabel 4.8 Tabel pengujian pelacakan kedua pada siang hari purwarupa M4

Percobaan	Lama Pelacakan (detik)
1	43
2	48

Tabel 4.9 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada siang hari purwarupa SIG552

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Warna putih yang digunakan sasaran memiliki kontras warna yang berbeda dengan <i>background</i> yang gelap akibat terkena bayangan. Estimasi jarak awal 10m, jarak akhir 6m</p>
2		<p>Sasaran bergerak dengan kecepatan yang lebih cepat dari percobaan 1. Warna putih yang digunakan sasaran memiliki kontras warna yang berbeda dengan <i>background</i> yang gelap akibat terkena bayangan. Estimasi jarak awal 11m, jarak akhir 9m</p>

Tabel 4.10 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada siang hari purwarupa M4

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Warna putih yang digunakan sasaran memiliki kontras warna yang berbeda dengan <i>background</i> yang gelap akibat terkena bayangan.</p>
2		<p>Sasaran bergerak dengan kecepatan yang sama dengan percobaan sebelumnya. Warna putih yang digunakan sasaran memiliki kontras warna yang berbeda dengan <i>background</i> yang gelap akibat terkena bayangan.</p>

Dari data-data pada pengujian pelacakan pertama diatas dapat disimpulkan bahwa kecepatan gerak dan perbedaan warna sasaran dengan *background* memiliki pengaruh terhadap proses pelacakan. Jika sasaran bergerak dengan cepat maka sasaran mudah lepas dari sistem karena perbedaan piksel terhadap waktu pada *image* yang tertangkap terlalu besar sehingga pelacakan sasaran dengan metode *optical flow* mudah lepas. Sedangkan untuk perbedaan warna antara sasaran dengan *background* memiliki pengaruh yang besar juga. Karena metode ini melihat pergeseran *flow* dari warna pada piksel, jika terdapat gradasi warna yang rendah maka posisi sasaran bisa berubah dan dapat lepas dari sasaran.

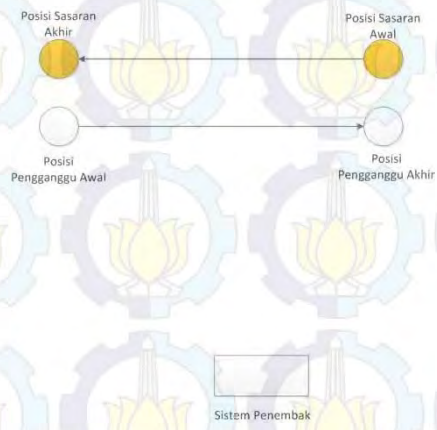
Pada pengujian pelacakan kedua, beban purwarupa yang digunakan memiliki pengaruh terhadap gerakan motor. Pada purwarupa SIG 552, pergerakan horizontal memiliki respon lebih cepat karena memerlukan tenaga gerak yang lebih kecil dari pada dengan beban purwarupa M4. Hal ini sesuai dengan pengujian motor pada pembahasan sebelumnya. Selain itu akibat dari beratnya beban M4, terjadi *overshoot* pada pergerakan motor dimana motor berhenti melebihi posisi yang ditentukan. Motor akan memperbaiki posisi kembali meskipun terjadi *overshoot*. Untuk pergerakan vertikal pada purwarupa M4 ditiadakan karena beratnya beban purwarupa menyebabkan pergerakan vertikal pada motor tidak memungkinkan.

Setelah dilakukan pengujian pelacakan, dilakukan pengujian lanjutan berupa gangguan pada sasaran yang telah dikunci dimana sasaran dilewati benda asing baik dibelakang atau didepan sasaran. Pengujian ini dilakukan pada kondisi cuaca cerah dan pencahayaan yang cukup di ruang terbuka.

- Waktu : Pukul 15.30 WIB
Tempat : Lapangan Parkir Dosen Teknik Elektro ITS
Ketinggian : 90cm

Pada pengujian ini diambil data apakah benda asing yang melewati sasaran yang telah dikunci baik melalui depan sasaran maupun belakang sasaran mempengaruhi penguncian sasaran apa tidak.

Tabel 4.11 Skenario pengujian gangguan sasaran

Percobaan	Alur Gangguan	Keterangan
1		<p>Sasaran diganggu dengan dilewati pengganggu di depannya sehingga sasaran sempat tertutup oleh pengganggu. Estimasi jarak sasaran 14m hingga 18m.</p>
2		<p>Sasaran diganggu dengan dilewati pengganggu di depannya sehingga pengganggu sempat tertutup oleh sasaran. Estimasi jarak sasaran 5m hingga 7m.</p>

Dari data yang didapat pada pengujian, didapat bahwa sasaran yang diganggu oleh pengganggu di depan sasaran mengalami gangguan pelacakan dan penguncian sasaran. Hal ini terjadi akibat sasaran tertutup oleh pengganggu yang melewatinya. Pada pengujian kedua dimana sasaran berada didepan pengganggu yang melewatinya, sistem tetap

mengunci sasaran tanpa adanya gangguan karena pengganggu yang melewati sasaran tidak menutupi sasaran.



Gambar 4.11 Pengujian gangguan di depan sasaran sebelum gangguan melewati sasaran



Gambar 4.12 Pengujian gangguan di depan sasaran sesudah gangguan menutupi sasaran

4.5 Pengujian Pelacakan dan Penguncian Sasaran di Malam Hari

Pada bagian ini dilakukan pengujian pada kondisi cahaya yang kurang terhadap sistem yang telah dibuat. Pada pengujian kali ini dilakukan dengan satu buah purwarupa senjata yaitu purwarupa SIG 552. Pengujian dilakukan pada malam hari dengan penerangan lampu. Pengujian dilakukan pada:

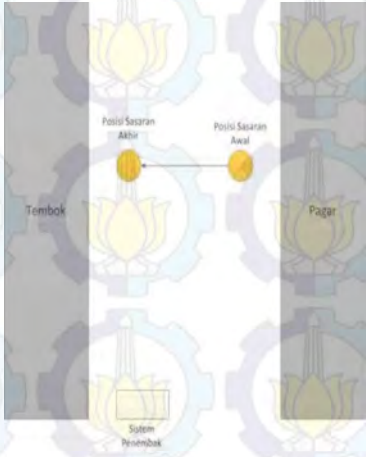
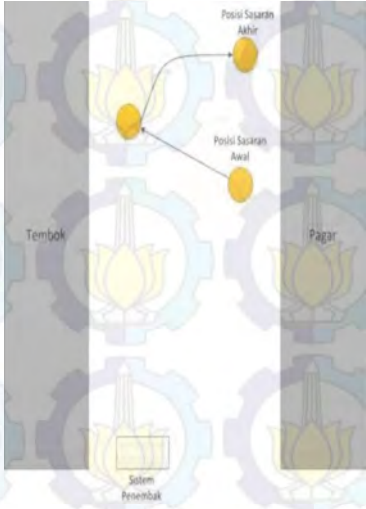
- Waktu : Pukul 19.00 WIB
Tempat : Koridor Lantai 2 Gedung B Jurusan Teknik Elektro ITS
Ketinggian : 90cm

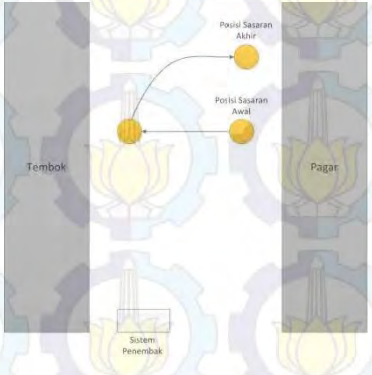
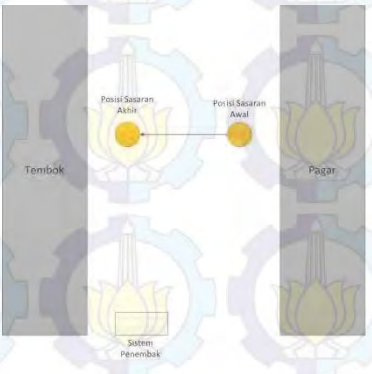
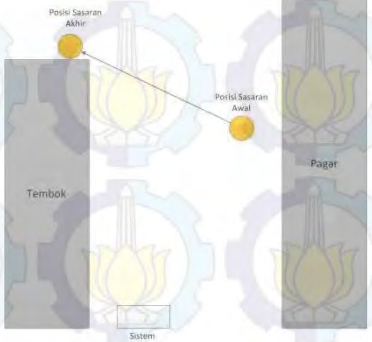
Pada pengujian ini dilakukan untuk mengetahui seberapa lama sistem dapat melacak dan mengunci sasaran dengan warna sasaran yang berbeda-beda namun dengan kondisi *background* dan posisi sistem senjata yang sama dan tidak dirubah. Selain itu pergerakan sasaran pada pengujian kali ini diatur memiliki jarak 4m hingga 5m saja. Kondisi pengujian dilakukan di lorong yang memiliki lebar area pergerakan sasaran sebesar 1,5m. Dari pengujian yang dilakukan tersebut didapat data-data sebagai berikut:

Tabel 4.12 Tabel pengujian pelacakan pada malam hari 1

Percobaan	Lama Pelacakan (detik)	warna sasaran
1	2	abu abu
2	5	abu abu
3	0	biru
4	2	merah
5	9	merah

Tabel 4.13 Tabel alur gerak sasaran dan keterangan pengujian pelacakan pada siang hari 1

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna belakangnya serta pencahayaan yang kurang, sehingga sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4m.</p>
2		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna belakangnya serta pencahayaan yang kurang, sehingga sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4-5m.</p>

3		<p>Sasaran bergerak secara lambat. Karena warna sasaran ada kesamaan dengan warna pagar serta pencahayaan yang kurang, sehingga sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4-5m.</p>
4		<p>Sasaran bergerak secara lambat. Warna sasaran memiliki kontras warna yang berbeda dengan warna belakangnya namun pencahayaan yang kurang menyebabkan sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4m.</p>
5		<p>Sasaran bergerak secara lambat. Warna sasaran memiliki kontras warna yang berbeda dengan warna belakangnya. Sasaran terlepas dari pelacakan karena tertutup oleh tembok. Estimasi jarak sasaran 4-5m.</p>

Selain itu dilakukan pengujian pelacakan kedua pada hari yang sama dengan kondisi pengujian dilakukan pada malam hari dengan penerangan lampu seperti pada kondisi pengujian malam sebelumnya. Namun dengan jenis senjata yang berbeda dan sasaran yang sama. Pengujian dilakukan pada:

- Waktu : Pukul 20.00 WIB
Tempat : Koridor Lantai 2 Gedung B Teknik Elektro ITS
Ketinggian : 90cm

Pengujian dilakukan seberapa lama sistem dapat melacak dan mengunci sasaran dengan sasaran yang sama namun dengan jenis purwarupa senjata yang berbeda. Pada pengujian kali ini dilakukan dengan 2 jenis purwarupa yang berbeda yaitu purwarupa SIG 552 dengan massa 0,5kg dan purwarupa M4 dengan massa 4kg. Pada pengujian ini sasaran yang digunakan menggunakan warna yang tetap yaitu warna merah. Dari pengujian tersebut didapat data-data sebagai berikut:

Tabel 4.14 Tabel pengujian pelacakan kedua pada malam hari purwarupa SIG552

Percobaan	Lama Pelacakan (detik)
1	2
2	9

Tabel 4.15 Tabel pengujian pelacakan kedua pada malam hari purwarupa M4

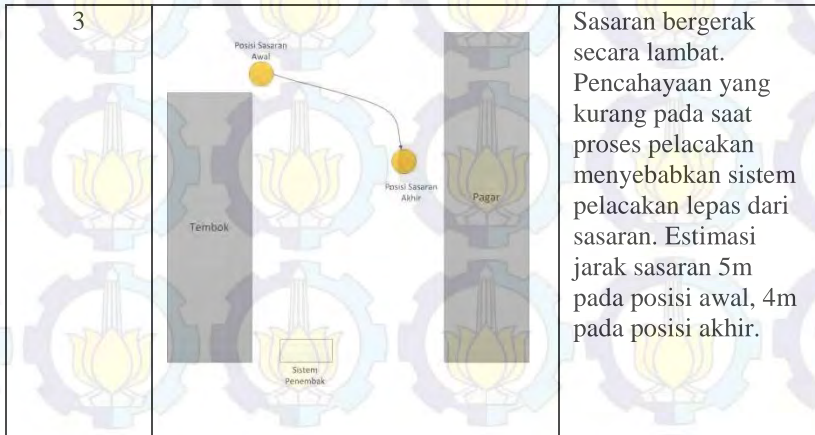
Percobaan	Lama Pelacakan (detik)
1	20
2	15
3	15

Tabel 4.16 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada malam hari purwarupa SIG552

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Pencahayaannya yang kurang pada saat proses pelacakan menyebabkan sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 4m.</p>
2		<p>Sasaran bergerak secara lambat. Pencahayaannya yang kurang pada saat proses pelacakan menyebabkan pelacakan mengalami gangguan. Sasaran terlepas dari pelacakan karena sasaran tertutup oleh tembok. Estimasi jarak sasaran 4-5m.</p>

Tabel 4.17 Tabel alur gerak sasaran dan keterangan pengujian pelacakan kedua pada malam hari purwarupa M4

Percobaan	Alur Pergerakan Sasaran	Keterangan
1		<p>Sasaran bergerak secara lambat. Pencapaian yang kurang pada saat proses pelacakan menyebabkan sistem pelacakan lepas dari sasaran. Estimasi jarak sasaran 5m pada posisi awal, 4m pada posisi akhir.</p>
2		<p>Sasaran bergerak secara lambat. Pencapaian yang kurang pada saat proses pelacakan menyebabkan pelacakan mengalami gangguan. Sasaran terlepas dari pelacakan karena sasaran tertutup oleh tembok. Estimasi jarak sasaran 4-5m.</p>



Gambar 4.13 Proses pengujian pelacakan pada malam hari

Dari data-data pada pengujian pelacakan pada malam hari yang pertama diatas dapat disimpulkan bahwa kondisi cahaya dan perbedaan warna sasaran dengan *background* memiliki pengaruh terhadap proses pelacakan. Cahaya yang kurang terang menyebabkan proses pengambilan gambar lambat, (*frame per detik*) sehingga menyebabkan

gerak motor mengalami kelebihan dan pergeseran *image* mengalami *blur*. Sedangkan untuk perbedaan warna antara sasaran dengan *background* memiliki pengaruh yang besar juga. Karena metode ini melihat pergeseran *flow* dari warna pada piksel, jika terdapat gradasi warna yang rendah maka posisi sasaran bisa berubah dan dapat lepas dari sasaran.

Pada pengujian pelacakan kedua, seperti pada peengujian malam hari yang pertama, kondisi cahaya mempengaruhi proses pelacakan. Selain itu beban purwarupa yang digunakan memiliki pengaruh terhadap gerakan motor. Pada purwarupa SIG 552, pergerakan horizontal memiliki respon lebih cepat karena memerlukan tenaga gerak yang lebih kecil dari pada dengan beban purwarupa M4. Hal ini sesuai dengan pengujian motor pada pembahasan sebelumnya. Selain itu akibat dari beratnya beban M4, terjadi *overshoot* pada pergerakan motor dimana motor berhenti melebihi posisi yang ditentukan. Motor akan memperbaiki posisi kembali meskipun terjadi *overshoot*. Untuk pergerakan vertikal pada purwarupa M4 ditiadakan karena beratnya beban purwarupa menyebabkan pergerakan vertikal pada motor tidak memungkinkan.

4.6 Pengujian Penembakan

Dengan menggunakan metode yang telah dijelaskan pada bagian perancangan sistem, maka dilakukan pengujian penembakan terhadap sistem yang telah dibuat. Pengujian bertujuan untuk menganalisis keakuratan penguncian sasaran pada sistem. Pada pengujian kali ini dilakukan dengan 1 jenis purwarupa yaitu purwarupa SIG 552. Sasaran yang digunakan memiliki diameter 20cm. Pengujian ini dilakukan pada ruang tertutup dengan pencahayaan yang cukup.

- Waktu : Pukul 19.30 WIB
Tempat : Laboratorium B202 Jurusan Teknik Elektro
Ketinggian : 120cm

Pengujian dilakukan dengan ketinggian sasaran dan jarak sasaran yang berbeda-beda. Sebelum dilakukan penembakan, sistem terlebih dahulu mengunci sasaran. Dari pengujian tersebut didapat data-data sebagai berikut:

Tabel 4.18 Tabel pengujian tembak.

Percobaan	Jarak Sasaran	Ketinggian Sasaran
1	5m	150cm
2	4m	150cm
3	3m	150cm
4	4m	120cm
5	5m	120cm
6	3m	120cm

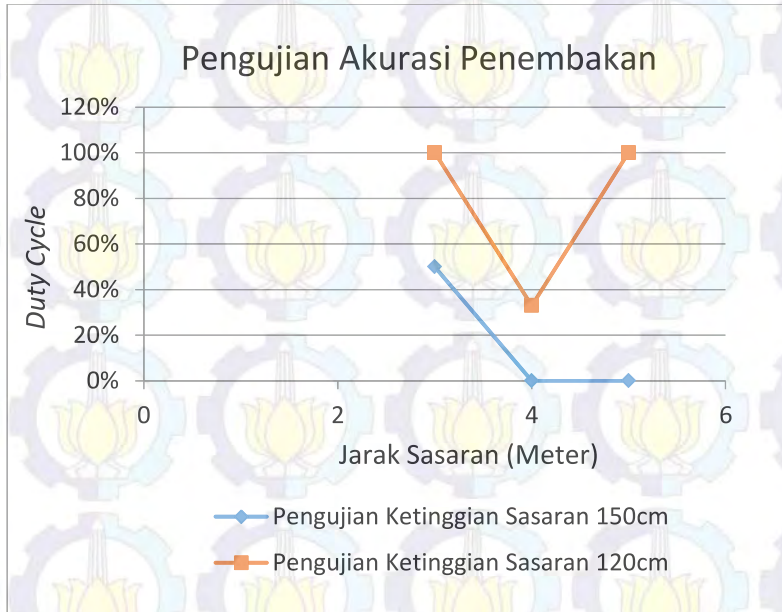
Tabel 4.19 Tabel hasil pengujian tembak

Percobaan	Peluru Ditembakkan	Peluru Kena Sasaran	Gambar
1	3	0	

2	3	0	
3	2	1	

4	3	1	
5	4	4	
6	4	4	

Dari data yang didapat dapat disimpulkan akurasi penembakan terhadap jarak dan ketinggian sasaran yaitu:



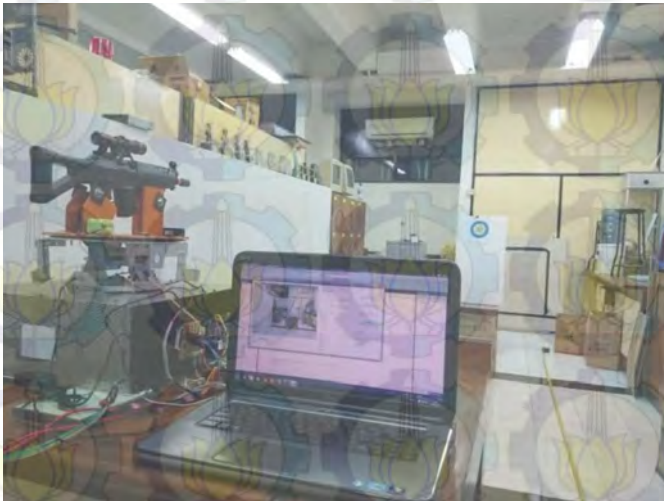
Gambar 4.14 Grafik Akurasi Penembakan Sasaran

Pada pengujian ini dianggap terkena sasaran jika peluru penembakan mengenai lingkaran dengan diameter 20cm tersebut (minimal lingkaran berwarna biru). Sasaran tersebut sesuai dengan standar dari ISSF (*International Shooting Sport Federation*) untuk sasaran berjarak 50 yard dengan senjata pistol.


Pada sistem dirancang dengan perhitungan gerakan GLBB dimana kecepatan least peluru dari sistem purwarupa SIG552 ini adalah kurang lebih 120kaki per detik dan dengan ketinggian sistem 120cm, jarak maksimum yang dapat ditempuh dengan ketinggian sasaran minimal 120cm adalah 20meter.

Dari data-data tersebut dapat dilihat bahwa ketinggian dan jarak sasaran mempengaruhi akurasi penembakan dimana saat sasaran memiliki ketinggian lebih dari ketinggian sistem, akurasi penembakan menurun. Selain itu semakin jauh jarak sasaran terhadap sistem juga mempengaruhi akurasi. Semakin jauh sasaran dari sistem penembak, akurasi penembakan semakin menurun.

Sistem penembakan ini adalah *openloop* sehingga kompensasi pada salah penembakan pada sistem tidak ada. Untuk diluar sistem, kesalahan penembakan dapat dituntut di pengadilan dengan sistem peradilan pidana. Di pengadilan tersebut akan ditentukan apakah ada suatu kompensasi ataupun hukuman akibat salah tembak atau tidak. [13]



Gambar 4.15 Proses pengujian penembakan



Halaman ini sengaja dikosongkan

BAB V

KESIMPULAN DAN SARAN

Setelah tahap pengujian sistem, dilakukan pengujian sistem untuk beberapa kondisi cahaya

Kesimpulan

Kesimpulan yang diperoleh dalam Tugas Akhir ini adalah :

1. Massa dari Senjata mempengaruhi respon sistem penggerak vertikal maupun horizontal. Semakin besar massa senjata, maka beban pada motor semakin besar dan menyebabkan respon motor melambat. Dengan massa purwarupa senjata yang diberikan adalah 0,5kg untuk SIG 552 dan 4kg untuk M4.
2. Performa sistem dengan metode *optical flow* untuk perhitungan pelacakan dengan *sampling time* setiap 40ms dan *frame* 480p tidak terjadi keterlambatan. *Sampling time* 40ms tersebut cukup untuk melakukan perhitungan dengan metode *optical flow* dengan spesifikasi komputer yang digunakan pada tugas akhir ini.
3. Hasil pengujian pelacakan pada sasaran didapat sasaran paling tidak memiliki gradasi warna yang berbeda dengan *background*. Sasaran memiliki perbedaan intensitas minimal 30 perubahan untuk kanal warna RGB 24 bit.
4. Hasil pengujian penembakan dengan standar sasaran ISSF untuk jarak 50yard memiliki rata-rata akurasi lebih dari 50% pada jarak sasaran tidak kurang dari 5m dengan sasaran memiliki ketinggian sejajar dengan sistem.

Saran

Beberapa saran yang penulis dapat berikan untuk pengembangan tugas akhir adalah sebagai berikut :

1. Sensor visual yang digunakan diharapkan lebih memiliki kualitas *Frame Per Second* (FPS) yang lebih baik dengan *frame per second* diatas 25FPS.

2. Diharapkan untuk menambah akurasi diberi perhitungan jenis peluru yang digunakan.

3. Demi mendapatkan proses yang lebih cepat dengan *Frame Per Second* (FPS) yang lebih baik, penggunaan *processing unit* dapat diganti dengan PC yang memiliki *processor* lebih cepat.



LAMPIRAN

Berikut adalah listing program dari tugas akhir saya,

```
Imports Emgu.CV
Imports Emgu.CV.CvEnum
Imports Emgu.CV.Util
Imports Emgu.CV.Structure
Imports Emgu.Util
Imports Emgu.CV.Features2D
Imports Emgu.CV.UI

Public Class Form1

    Dim Tengah As Rectangle

    Dim manualH As Double
    Dim manualL As Double
    Dim AO0 As New InstantAoCtrl
    Dim AO1 As New InstantAoCtrl
    Dim AI0 As New InstantAiCtrl
    Dim AI1 As New InstantAiCtrl
    Dim DigitalOut As New InstantDoCtrl
    Dim DataOut0 As Double
    Dim DataOut1 As Double
    Dim DataOutD As Double
    Dim NILAIHP As Double
    Dim NILAIVP As Double
    Dim NILAIVD As Double
    Dim NILAIHD As Double
    Dim NILAIHI As Double
    Dim NILAIVI As Double
    Dim DELTAERRORV As Double
    Dim DELTAERRORH As Double
    Dim PWMV As Double
    Dim PWMH As Double
    Dim XP As Double
    Dim YP As Double
    Dim YD As Double
    Dim XD As Double
    Dim XI As Double
```

```

Dim YI As Double
Dim Minus As Double
Dim DataIn1 As Double
Dim DataIn2 As Double

Dim tembak As Integer
Dim a As Integer
Dim b As Integer
Dim c As Integer
Dim d As Integer
Dim hasil As Integer
Dim errorx As Integer
Dim errory As Integer
Dim errorx1 As Integer
Dim errory1 As Integer
Dim errorxi As Integer
Dim erroryi As Integer
Dim titikengahx As Integer
Dim titikengahy As Integer

Dim xx As Integer
Dim yy As Integer

Dim capdev As New Capture(1)
Dim citraasli As New Image(Of Bgr, Byte)(640, 480)
Dim imColor As New Image(Of Bgr, Byte)(640, 480)
Dim imGray As New Image(Of Gray, Byte)(640, 480)
Dim imGray2 As New Image(Of Gray, Byte)(640, 480)
Dim bunder As Rectangle
Dim roi_wajah As Rectangle

Dim face As New
HaarCascade("haarcascade_upperbody.xml")
Dim faceDetected As MCvAvgComp()
Dim thershold As Double

Private captureInProgress As Boolean
Const radius = 5

```



```

Dim defineCorners As Boolean = False
Dim puntiA(0)() As PointF
Dim puntiB(0)() As PointF
Dim puntiTemp As PointF()()
Dim puntiBtemp(0)() As PointF
Dim add_remove_pt As Boolean = False
Dim count As Integer = 0
Dim lkcount, plk1, plk2, plk3, plk4, plk5, delplk As
Integer
Dim frame As Emgu.CV.Image(Of Emgu.CV.Structure.Bgr,
Byte)
Dim grayA As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim grayB As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim pyrBufferA As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim pyrBufferB As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim imgTemp As Emgu.CV.Image(Of
Emgu.CV.Structure.Gray, Byte)
Dim flags As Emgu.CV.CvEnum.LKFLOW_TYPE =
Emgu.CV.CvEnum.LKFLOW_TYPE.DEFAULT
Dim status As Byte()
Dim errors As Single()

Dim posisiX, posisiY, dposisiX, dposisiY, pposisiX,
pposisiY As Integer
Dim clickcount, klik_status, flag_lk As Integer

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
 MyBase.Load

capdev.SetCaptureProperty(CAP_PROP.CV_CAP_PROP_FRAME_WID
TH, 640)

capdev.SetCaptureProperty(CAP_PROP.CV_CAP_PROP_FRAME_HEI
GHT, 480)
        flag_lk = 0

```

```

manualH = 2
manualL = 2
AO0.SelectedDevice = New DeviceInformation(1)
AO1.SelectedDevice = New DeviceInformation(1)
DigitalOut.SelectedDevice = New
DeviceInformation(1)
AI0.SelectedDevice = New DeviceInformation(1)
AI1.SelectedDevice = New DeviceInformation(1)

a = 0
b = 0
c = 0
d = 0
titiktengahx = 320
titiktengahy = 240
tembak = 0
YI = 0
YD = 0
XD = 0
XI = 0

'Timer1.Enabled = True
End Sub

Private Sub Timer1_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
    citraasli = capdev.QueryFrame
    imColor = citraasli

    Tengah = New Rectangle(310, 230, 10, 10)
    citraasli.Draw(Tengah, New Bgr(Color.Black), 3)
    lukas()
    FaceDetection()
    PictureBox1.Image = citraasli.ToBitmap

    TextBox15.Text = dposisiX
    TextBox16.Text = dposisiY

```



```
TextBox17.Text = pposisiX  
TextBox18.Text = pposisiY
```

```
TextBox1.Text = errorx  
TextBox2.Text = errory  
TextBox3.Text = a  
TextBox4.Text = b  
TextBox5.Text = c  
TextBox6.Text = d  
TextBox7.Text = posisiX  
TextBox8.Text = posisiY
```

```
errorx = posisiX - 320  
errory = posisiY - 240
```

```
AI0.Read(0, DataIn1)  
AI1.Read(1, DataIn2)  
TextBox9.Text = DataIn1
```

```
    If DataIn2 > 2 Then  
        'proximity Horizontal  
        a = a  
        b = b  
        c = 0  
        d = 0
```

```
    End If
```

```
    If errorx < 32 Then  
        'kanan dan kiri berhenti  
        a = a  
        b = b  
        c = 0  
        d = 0
```

```
    End If
```

```

    If errorx < 24 Then
        bawah berhenti
        a = 0
        b = 0
        c = c
        d = d
    End If

    If errorx < -24 Then
        atas
        a = 0
        b = 1
        c = c
        d = d
    End If

    If errorx > 24 Then
        bawah
        a = 1
        b = 0
        c = c
        d = d
    End If

    If errorx < -32 Then
        kanan
        a = a
        b = b
        c = 1
        d = 0
    End If

    If errorx > 32 Then
        kiri
        a = a
        b = b
        c = 0
        d = 1
    End If

```

```

        If errorx = -320 And errory = -240 Then
'berhenti total, over pixel

            a = 0
            b = 0
            c = 0
            d = 0
        End If

        If DataIn1 < 0.5 Then                                'proximity
Vertikal
            a = a
            b = b
            c = 0
            d = 0
        End If

        hasil = 8 * d + 4 * c + 2 * b + 1 * a
        DataOutD = 16 * tembak + hasil
        If hasil = 0 Then
'berhenti
            ' DataOutD = 0
            DigitalOut.Write(0, DataOutD)
        End If
        If hasil = 1 Then
'bawah
            'DataOutD = 1
            DigitalOut.Write(0, DataOutD)
        End If
        If hasil = 2 Then                                    'atas
            'DataOutD = 2
            DigitalOut.Write(0, DataOutD)
        End If
        If hasil = 4 Then                                    'kiri
            'DataOutD = 4

```

```

        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 8 Then
'kanan
        'DataOutD = 8
        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 5 Then
'atas
        'DataOutD = 5
        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 6 Then
'bawah
        'DataOutD = 6
        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 9 Then
'kanan atas
        ' DataOutD = 9
        DigitalOut.Write(0, DataOutD)
    End If
    If hasil = 10 Then
'kanan bawah
        ' DataOutD = 10
        DigitalOut.Write(0, DataOutD)
    End If

    'Fungsi PWM
    If errorx < 0 Then
        XP = (-1 * errorx)
    End If
    If errorx > 0 Then
        XP = errorx
    End If
    If errory < 0 Then
        YP = (-1 * errory)
    End If
    If errory > 0 Then
        YP = errory
    End If

```

```

End If

NILAIVP = ((YP / 240) * 0.6) 'Control
P Vertikal
If (YP / 240) > 0.9999 Then
    NILAIVP = 0
End If
DELTAERRORV = ((YP / 240) - (YD / 240))
If DELTAERRORV < 0 Then
    DELTAERRORV = (-1 * DELTAERRORV)
End If
NILAIVD = ((DELTAERRORV / 40) * 24) 'Control
D Vertikal
YD = YP

If errorry > -240 Then
    errorryi = errorry
End If
If errorry = -240 Then
    errorryi = 0
End If
YI = YI + (errorryi / 240)
'Control I Vertikal

If YP < 24 Then
    YI = 0
End If
NILAIVI = (YI / 40) * 1.4
If NILAIVI < 0 Then
    NILAIVI = NILAIVI * -1
End If

PWMV = NILAIVP + NILAIVD + NILAIVI + 0.6 '
'Control PID Vertikal
If PWMV > 3 Then
    PWMV = 3

```


End If

TextBox11.Text = PWMV
A00.Write(0, PWMV)

P Horizontal NILAIHP = ((XP / 320) * 0.4) 'Control

If (XP / 320) > 0.9999 Then
 NILAIHP = 0

End If

DELTAERRORH = ((XP / 320) - (XD / 320))

If DELTAERRORH < 0 Then
 DELTAERRORH = (-1 * DELTAERRORH)

End If

'Control D Horizontal
NILAIHD = ((DELTAERRORH / 40) * 32)

XD = XP

If errorx > -320 Then
 errorxi = errorx

End If

If errorx = -320 Then
 errorxi = 0

End If

XI = XI + (errorxi / 320)

'Control I Horizontal

If XP < 32 Then
 XI = 0

End If

NILAIHI = (XI / 40) * 2

If NILAIHI < 0 Then
 NILAIHI = NILAIHI * -1

End If


```

        PWMH = NILAIHP + NILAIHD + NILAIHI + 1.2
'Control PID Horizontal
    If PWMH > 3 Then
        PWMH = 3
    End If

    TextBox12.Text = PWMH
    AO1.Write(1, PWMH)

    If DataIn1 < 0.5 Then 'proximity
Vertikal
        a = a
        b = b
        c = 0
        d = 0
    End If

    TextBox15.Text = NILAIVP
    TextBox16.Text = NILAIVD
    TextBox17.Text = NILAIVI
    TextBox18.Text = NILAIHP
    TextBox19.Text = NILAIHD
    TextBox20.Text = NILAIHI

End Sub
Private Sub tambah_titik(ByVal input_x, ByVal
input_y)
    Dim removedpoint As Boolean = False
    Dim radiussquare As Double = radius ^ 2
    removedpoint = False

    'menambah titik baru...
    If Not removedpoint Then
        puntiBtemp(0) = New PointF() {New
PointF(input_x, input_y)}
        add_remove_pt = True
    End If

```

```

End Sub
Private Sub lukas()

    frame = citraasli
    grayB = frame.Convert(Of Emgu.CV.Structure.Gray,
Byte)()

    If defineCorners Then
        count = 500
        puntiB = grayB.GoodFeaturesToTrack(count,
0.01, 10, 3)
        grayB.FindCornerSubPix(puntiB, New
System.Drawing.Size(10, 10), New System.Drawing.Size(-1,
-1), New Emgu.CV.Structure.MCvTermCriteria(20, 0.03))
        count = puntiB(0).Length
        defineCorners = False
    End If

    If count > 0 Then

        Emgu.CV.OpticalFlow.PyrLK(grayA, grayB,
pyrBufferA, pyrBufferB, puntiA(0), New Size(10, 10), 5,
New Emgu.CV.Structure.MCvTermCriteria(20, 0.03D), flags,
puntiB(0), status, errors)
        flags =
Emgu.CV.CvEnum.LKFLOW_TYPE.CV_LKFLOW_PYR_A_READY

        Dim k As Integer = 0
        Dim col2 As New Emgu.CV.Structure.Bgr(255,
128, 0)

        For i As Integer = 0 To count - 1

            If status(i) > 0 Then

                Dim cr As New
Emgu.CV.Structure.CircleF(puntiB(0)(i), 4)
                puntiB(0)(k) = puntiB(0)(i)
                frame.Draw(cr, col2, 2)
            End If
        Next i
    End If
End Sub

```

k += 1

```
End If
Next

count = k
posisiX = Int(puntiB(0)(0).X)
posisiY = Int(puntiB(0)(0).Y)
errorx1 = posisiX - 320
errory1 = posisiY - 240
End If

If add_remove_pt Then
    ' grayB.FindCornerSubPix(puntiBtemp, New
    System.Drawing.Size(10, 10), New System.Drawing.Size(-1,
    -1), New Emgu.CV.Structure.MCvTermCriteria(20, 0.03))

    If count > 0 Then
        ReDim Preserve
        puntiB(0)(puntiB(0).Length)
        puntiB(0)(puntiB(0).Length - 1) =
        puntiBtemp(0)(0)
    Else
        puntiB(0) = puntiBtemp(0)
    End If

    count = puntiB(0).Length
    add_remove_pt = False
End If

' swap images:
imgTemp = grayA
grayA = grayB
grayB = imgTemp
'swap pyramid
imgTemp = pyrBufferA
pyrBufferA = pyrBufferB
pyrBufferB = imgTemp
' swap points:
```

```

puntiTemp = puntiA
puntiA = puntiB
puntiB = puntiTemp
End Sub
Private Sub FaceDetection()

    imGray = citraasli.Convert(Of Gray, Byte)()

    faceDetected = imGray.DetectHaarCascade(face,
1,1, 3, CvEnum.HAAR_DETECTION_TYPE.DO_ROUGH_SEARCH, New
Size(30, 30))()

    xx = 0
    yy = 0
    For Each fo In faceDetected
        imGray.ROI = fo.rect
        xx = (fo.rect.X + (fo.rect.Width / 2))
        yy = (fo.rect.Y + (fo.rect.Height / 2))
        roi_wajah = New Rectangle(fo.rect.X,
fo.rect.Y, ((fo.rect.Width)), fo.rect.Height)
        bunder = New Rectangle((fo.rect.X +
(fo.rect.Width / 2)), (fo.rect.Y + (fo.rect.Height /
2))), 5, 5)
        imGray2.ROI = bunder
        imGray.ROI = roi_wajah
        citraasli.Draw(fo.rect, New Bgr(Color.Red),
1)
        citraasli.Draw(bunder, New Bgr(Color.Aqua),
2)
        imGray.ROI = Rectangle.Empty
        imGray2.ROI = Rectangle.Empty
    Next
End Sub
Private Sub PictureBox1_MouseDown(ByVal sender As
Object, ByVal e As System.Windows.Forms.MouseEventArgs)
Handles PictureBox1.MouseDown

    If e.Button = MouseButton.Left Then
        tambah_titik(e.X, e.Y)
    End If

```



```

End Sub

Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
    Application.Exit()
End Sub

Private Sub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BUTTON2.Click
    If BUTTON2.Text = "START" Then

capdev.SetCaptureProperty(CvEnum.CAP_PROP.CV_CAP_PROP_FR
AME_WIDTH, 640)

capdev.SetCaptureProperty(CvEnum.CAP_PROP.CV_CAP_PROP_FR
AME_WIDTH, 480)

        Timer1.Enabled = True
        BUTTON2.Text = "STOP"
    Else
        Timer1.Enabled = False
        BUTTON2.Text = "START"
    End If
End Sub

Private Sub Button3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button3.Click

    a = 0
    b = 0
    c = 0
    d = 0
    NILAIVP = 0
    NILAIVD = 0

```

```

NILAIVI = 0
NILAIHP = 0
NILAIHD = 0
NILAIHI = 0

PWMH = 0
PWMV = 0
XP = 0
XI = 0
YP = 0
YI = 0

error1 = 0
errorx1 = 0
errorx = 0
error1 = 0
posisiX = 0
posisiY = 0

For i As Integer = 0 To count - 1

    If status(i) > 0 Then

        Dim cr As New
        Emgu.CV.Structure.CircleF(puntiB(0)(i), 4)

        puntiB(0)(i).X = 0
        puntiB(0)(i).Y = 0

    End If
Next
count = 0
End Sub

Private Sub Button4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
    If tembak = 0 Then
        Button4.Text = "BERHENTI"
        tembak = 1
    End If
End Sub

```



```

Else
    tembak = 0
    Button4.Text = "TEMBAK"
End If

End Sub

Private Sub Button5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
    AO0.Write(0, manualH)
    DigitalOut.Write(0, 2)
End Sub

Private Sub Button8_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
    AO0.Write(0, 0)
    AO1.Write(1, 0)
    DigitalOut.Write(0, 0)

End Sub

Private Sub Button6_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button6.Click
    AO1.Write(1, manualL)
    DigitalOut.Write(0, 4)
End Sub

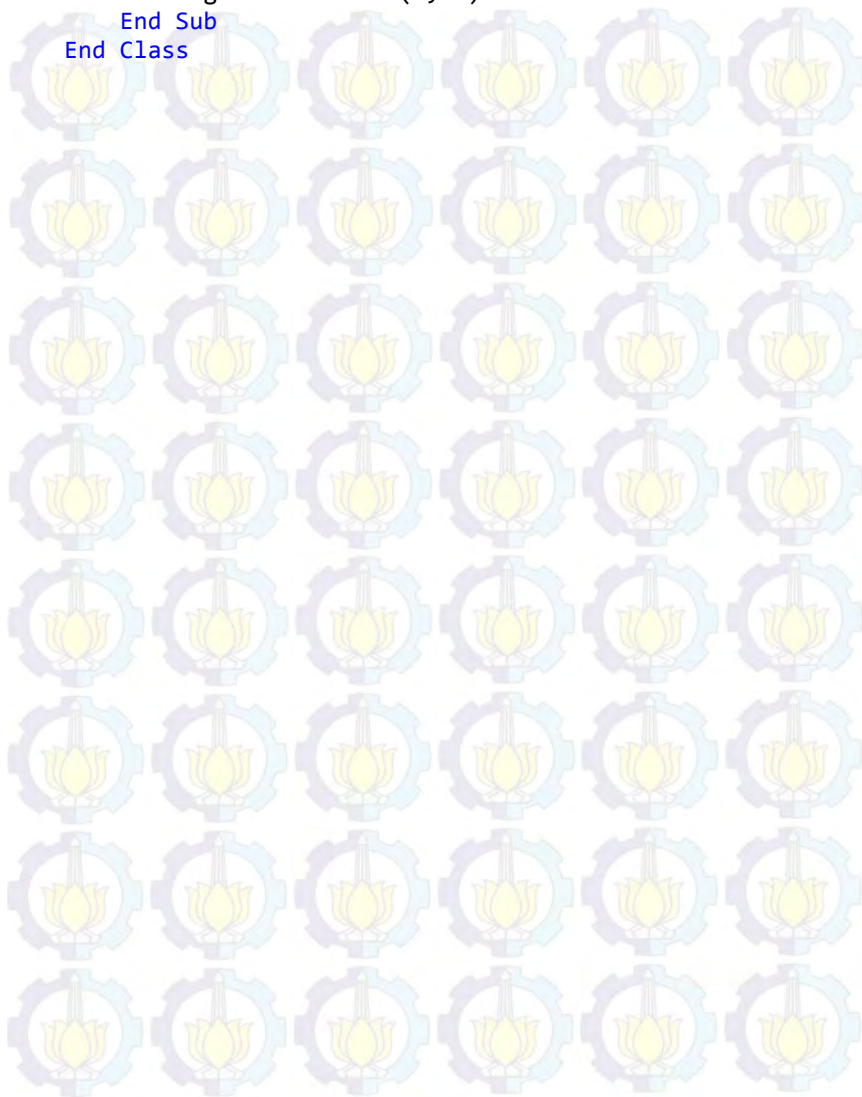
Private Sub Button9_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button9.Click
    AO0.Write(0, manualH)
    DigitalOut.Write(0, 1)
End Sub

Private Sub Button7_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button7.Click

```

```
A01.Write(1, manualL)  
DigitalOut.Write(0, 8)
```

```
End Sub  
End Class
```



DAFTAR PUSTAKA

- [1] Johnny Andrean Susanto, Kelvin Giovanni Lukman, "Digital Watermarking Untuk Melindungi Informasi Multimedia Dengan Metode Fast Fourier Transform (FFT)", <URL:http://eprints.mdp.ac.id/891/1/JURNAL%202009250032%20JOHNNY_ANDREAN_SUSANTO%20DAN%202009250027%20KELVIN_GIOVANNI_LUKMAN.pdf>,2003.
- [2] _____, "Teori dasar citra digital",<URL:<http://repository.usu.ac.id/bitstream/123456789/16877/4/Chapter%20II.pdf>>
- [3] Putra, Darma, "Pengolahan Citra Digital", Yogyakarta, Penerbit ANDI, 2010.
- [4] Rachmawati, "Estimasi Parameter Geometris Benda Berbasis Pengolahan Citra Digital", <URL: http://digilib.telkomuniversity.ac.id/index.php?option=com_content&view=article&id=344:citra-digital&catid=15:pemrosesan-sinyal&Itemid=14>, Desember, 2008.
- [5] Achmad Rizal, "Pengolahan Citra", <URL: <http://achmadrizal.staff.telkomuniversity.ac.id/2014/06/19/pengolahan-citra/>>, Juni, 2014.
- [6] Ubaidillah Umar, Reni Soelistijorini, Haryadi Amran Darwito, "Tracking Arah Gerakan Telunjuk Jari Berbasis Webcam Menggunakan Metode Optical Flow", Electronic Engineering Polytechnic Institute of Surabaya (EEPIS), Indonesia, October 26, 2011.
- [7] Indra Pramana, M Zen Hadi Samson, Setiawardhana, "TRACKING OBJECT MENGGUNAKAN METODE TEMPLATE MATCHING BERBASIS STEREO VISION", Surabaya.
- [8] Bradski, Gary dan Kaebler, Adrian. "*Learning OpenCV Computer Vision with the OpenCV Library*", O' REILLY. 2008.
- [9] _____, "Advantech USB-4711A Module", <URL: http://www2.advantech.eu/products/1-2MLKNO/USB4711A/mod_16CB666D-62EE-4C15-BF5F-EA49F61093CF.aspx>

- [10] Williams, Al “*Microcontroller projects using the Basic Stamp*” Edisi kedua, *Focal Press*.
- [11] Petty, Dan. “The US Navy – Fact File”, Navy.mil, 2013
- [12] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001, pp. I-511-I-518 vol.1.
- [13] Edi, Setiadi. ”*Kontroversi Pelaksanaan KUHAP*”, Pikiran Rakyat. 8 Pebruari 2003.

BIODATA PENULIS



Dani Prasetyawan lahir di Jombang pada 10 Agustus 1992. Anak pertama dari 2 bersaudara dari pasangan Djoko Purwanto dan Rivana Harijani. Penulis menyelesaikan pendidikan dasar di SDN Penjaringan Sari II Surabaya, Surabaya dilanjutkan dengan pendidikan menengah di SMPN 1 Surabaya dan SMAN 5 Surabaya. Pada tahun 2011, penulis memulai pendidikan di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi

Sepuluh Nopember (ITS) Surabaya. Selama kuliah penulis aktif membantu penyelenggaraan kegiatan dan aktif sebagai asisten laboratorium Elektronika Dasar dan praktikum Elektronika maupun koordinator asisten pada semester ganjildan genap 2014-2015.

Email :

daniprasetyawan1992@gmail.com

Halaman ini sengaja dikosongkan